



การใช้แมพรีดิคชันเชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบี

นายนิรุทธ์ รวยริน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีการศึกษา 2557

การใช้แมพริควิซซ์เชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบี

นายนิรุทธ์ รวยริน วท.บ. (วิทยาการคอมพิวเตอร์)

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2557

คณะกรรมการสอบวิทยานิพนธ์



ประธานกรรมการสอบวิทยานิพนธ์

(รศ. ดร.พีระพันธ์ โสพัศสถิตย์)



กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์

(ผศ. ดร.เกรียงไกร ปอแก้ว)



กรรมการ

(ผศ. ดร.สุริย์ ฟูนิลกุล)



กรรมการ

(ดร.อุมามพร สุภสิทธิเมธี)

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หัวข้อวิทยานิพนธ์	การใช้แมพรีดิคชันเชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบี
หน่วยกิต	12
ผู้เขียน	นายนิรุทธ์ รวยริน
อาจารย์ที่ปรึกษา	ผศ. ดร.เกรียงไกร ปอแก้ว
หลักสูตร	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
คณะ	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2557

บทคัดย่อ

ในปัจจุบัน ข้อมูลมีปริมาณเพิ่มมากขึ้นอย่างรวดเร็ว และมีความหลากหลายในเชิงโครงสร้าง จึงได้มีการนำฐานข้อมูลโนเอสคิวแอลซึ่งมีความสามารถในการจัดการกับข้อมูลปริมาณมากและมีความหลากหลายและยืดหยุ่นในเชิงโครงสร้างมาใช้ในการจัดการข้อมูล แต่ฐานข้อมูลประเภทนี้ มักไม่มีคำสั่งในการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลที่อยู่ต่างโครงสร้างกัน งานวิจัยนี้จึงได้ศึกษาและนำเสนอวิธีการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลที่อยู่ต่างโครงสร้างกันในฐานข้อมูลโนเอสคิวแอลโดยใช้มองโกดีบีและวิธีการเขียนโปรแกรมสำหรับจัดการข้อมูลด้วยแมพรีดิคชันเป็นกรณีศึกษาจากการศึกษา พบว่าการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลที่อยู่ต่างโครงสร้างกันโดยใช้แมพรีดิคชันทำให้สามารถเข้าถึงข้อมูลที่ต้องการได้อย่างมีประสิทธิภาพกว่าวิธีการอ่านข้อมูลจากแต่ละโครงสร้างออกมาจากฐานข้อมูลเพื่อเชื่อมโยงข้อมูลภายนอกฐานข้อมูล นอกจากนี้ ยังพบว่าทิศทางในการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลที่อยู่ต่างโครงสร้างกันด้วยแมพรีดิคชันมีผลต่อประสิทธิภาพในการทำงาน จากผลการทดลอง สามารถสรุปได้ว่า การเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลที่อยู่ต่างโครงสร้างกัน ควรเชื่อมโยงจากกลุ่มข้อมูลที่เหลือข้อมูลจากการกรองอยู่น้อยไปหากกลุ่มข้อมูลที่ไม่สามารถกรองข้อมูลได้หรือกรองข้อมูลออกได้น้อย

คำสำคัญ : เชื่อมคอลเลกชัน / โนเอสคิวแอล / มองโกดีบี / แมพรีดิคชัน

Thesis Title	The Use of MapReduce for joint Collection of MongoDB Database
Thesis Credits	12
Candidate	Mr. Nirut Ruayruen
Thesis Advisor	Asst. Prof. Dr. Kriengkrai Porkaew
Program	Master of Science
Field of Study	Information Technology
Faculty	School of Information Technology
Academic Year	2014

Abstract

At present, the volume of data is increasing rapidly and diverse in structure. It has adopted NoSQL database, due to its ability to handle large data volumes and diverse and flexible structure used to manage data. However, this database type lacks a command to connect different data structures. This research has studied and presented the relations between the different data structures in NoSQL database using MongoDB and programming methods for handling data formats with MapReduce in case studies. The study revealed that the joint connection between different data structures using MapReduce can make the information they need more accessible and efficient than reading the data of each structure from the database to connect data in external databases. It further found that the direction of the relations between the different structures with MapReduce affects the performance. From the study, it can be concluded that the joint connection between different data structures should be done from the smallest remained data which are filtered to the data can not be filtered or filtered out less.

Keywords : Join Collection / MapReduce / MongoDB/ NoSQL

กิตติกรรมประกาศ

ข้าพเจ้าในฐานะผู้วิจัยขอกราบขอบพระคุณ ผศ.ดร.เกรียงไกร ป่อแก้ว อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้สละเวลาให้ความรู้ คำปรึกษาพร้อมทั้งข้อเสนอแนะต่างๆ ที่เป็นประโยชน์กับงานวิจัย ซึ่งส่งผลให้งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณ รศ.ดร.พีระพนธ์ โสพัศสถิต ผศ.ดร.สุรีย์ ฟูนิลกุล และ ดร.อุมาพร สุภสิทธิเมธี ที่ท่านได้กรุณาเป็นคณะกรรมการในการสอบหัวข้อวิทยานิพนธ์และสอบวิทยานิพนธ์ขั้นสุดท้าย ผู้วิจัยขอขอบคุณ คุณศิริรัตนา นิลน้อย เป็นอย่างมากที่ช่วยให้คำแนะนำ คอยเป็นกำลังใจและสนับสนุนในการวิจัยครั้งนี้

ผู้วิจัยขอกราบขอบพระคุณอย่างสูงต่อคุณบิดา มารดา ที่ได้มอบชีวิต และอบรมเลี้ยงดู สนับสนุนให้ความช่วยเหลือผู้วิจัยในทุกๆด้านโดยตลอด สุดท้ายขอขอบคุณครูบาอาจารย์ทุกท่านที่ให้วิชาความรู้ ญาติมิตรเพื่อนทุกท่านที่ให้การสนับสนุนช่วยเหลือ ความดีอันใดที่เกิดจากงานวิจัยในครั้งนี้ ผู้วิจัยขออุทิศให้แก่บุคคลผู้มีพระคุณดังกล่าวทุกท่าน

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
รายการตาราง	ฉ
รายการรูปประกอบ	ช

บทที่

1. บทนำ	1
1.1 ความสำคัญและที่มาของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	3
1.3 ประโยชน์และผลที่คาดว่าจะได้รับจากงานวิจัย	3
1.4 ขอบเขตงานวิจัย	3
1.5 ขั้นตอนการดำเนินงาน	4
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 ความรู้พื้นฐานที่เกี่ยวข้อง	5
2.1.1 ฐานข้อมูล โนเอสคิวแอล (NoSQL Database)	5
2.1.2 มองโกดีบี (MongoDB)	10
2.1.3 แมพรีดิวซ์ (MapReduce)	13
2.2 งานวิจัยที่เกี่ยวข้อง	17
2.2.1 SimpleSQL: A Relational Layer for SimpleDB	17
2.2.2 Scalable Join Queries in Cloud Data Stores	18
2.2.3 งานวิจัยที่ประยุกต์ใช้ MapReduce กับ SPARQL	20
3. การใช้แมพรีดิวซ์เชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบี	23
3.1 รูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลกชัน	23
3.2 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง	25

3.3	รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แม่พรีดิคต์	28
3.3.1	ข้อกำหนดของไลบรารีการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แม่พรีดิคต์	30
3.3.2	ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แม่พรีดิคต์	31
3.4	รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แม่พรีดิคต์ร่วมกับ การสร้างคอลเลกชันชั่วคราว	32
3.4.1	ข้อกำหนดของไลบรารีการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้ แม่พรีดิคต์ร่วมกับการสร้างคอลเลกชันชั่วคราว	33
3.4.2	ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แม่พรีดิคต์ ร่วมกับการสร้างคอลเลกชันชั่วคราว	33
4.	ระเบียบวิธีการวิจัย	35
4.1	วิธีการทดลองและการประเมินผล	35
4.2	การวัดประสิทธิภาพค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไปคอลเลกชัน B	47
4.3	การวัดประสิทธิภาพค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A	49
4.4	การเปรียบเทียบค่าเวลาการทำรายการ	51
5.	บทสรุป	54
5.1	บทสรุปของงานวิจัย	55
5.2	แนวทางที่พัฒนาในอนาคตและข้อเสนอแนะ	56
	เอกสารอ้างอิง	57
	ประวัติผู้วิจัย	60

รายการตาราง

ตาราง		หน้า
4.1	แสดงจำนวนรายการข้อมูลของแต่ละเงื่อนไขที่ใช้ในการทดสอบการทำงานของไลบรารี	38
4.2	แสดงจำนวนรายการข้อมูลของแต่ละเงื่อนไขที่ใช้ในการทดลอง	38
4.3	ค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไปคอลเลกชัน B	48
4.4	ค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A	50

รายการรูปประกอบ

รูป	หน้า
2.1 ตัวอย่างการจัดเก็บข้อมูลแบบ Key-Value Data Model	6
2.2 ตัวอย่างการจัดเก็บข้อมูลแบบ Document Data Model	7
2.3 ตัวอย่างการจัดเก็บข้อมูลแบบ Column-Family Data Model	8
2.4 ตัวอย่างการจัดเก็บข้อมูลแบบ Graph Data Model	9
2.5 หลักการเก็บข้อมูลของฐานข้อมูล MongoDB กับข้อมูลของฐานข้อมูลเชิงสัมพันธ์	11
2.6 ตัวอย่างการจัดเก็บข้อมูลของฐานข้อมูล MongoDB	12
2.7 MapReduce Model Example Word Counting	14
2.8 รูปแบบของฟังก์ชันแมพและฟังก์ชันรีดิวซ์	15
2.9 map-reduce operation	16
2.10 Interface of SimpleSQL	18
2.11 CloudTPS's representation of a join query	19
2.12 Sample of RDF structure	20
2.13 Sample query of SPARQL	21
2.14 ตารางพื้นฐานที่ถูกสร้างขึ้นจากเงื่อนไข	21
3.1 รูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลกชัน	24
3.2 แสดงอัลกอริทึมเชื่อมคอลเลกชันจากการเขียน โปรแกรม โดยตรง	26
3.3 แสดงการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียน โปรแกรม โดยตรง	27
3.4 อัลกอริทึมเชื่อมคอลเลกชันจากการประยุกต์ใช้เทคนิคแมพรีดิวซ์	28
3.5 ข้อกำหนดของไลบรารีเพื่อเชื่อมความสัมพันธ์ของคอลเลกชัน	30
3.6 ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์	31
3.7 การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว	33
4.1 โครงสร้างของคอลเลกชันนักเรียนกับอาจารย์ที่ปรึกษา	36
4.2 สถานการณ์ในการทดลองเพื่อสอบถามข้อมูล	37
4.3 รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join	39
4.4 หน้าเว็บไซต์จำลองการเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join	40
4.5 แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join เสร็จสิ้น	40
4.6 รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join	41

รายการรูปประกอบ (ต่อ)

รูป	หน้า	
4.7	หน้าเว็บไซค์จำลองการรับพารามิเตอร์เพื่อเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join	42
4.8	แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join เสร็จสิ้น	43
4.9	แสดงผลการเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join	43
4.10	รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join	44
4.11	หน้าเว็บไซค์จำลองการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join	45
4.12	แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join เสร็จสิ้น	46
4.13	แสดงผลการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join	46
4.14	กราฟแสดงค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไปคอลเลกชัน B	48
4.15	กราฟแสดงค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A	50
4.16	กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีการเชื่อมคอลเลกชันรูปแบบ Direct join	51
4.17	กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีการเชื่อมคอลเลกชันรูปแบบ MR join	52
4.18	กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีเชื่อมคอลเลกชันรูปแบบ MR temp join	53

บทที่ 1 บทนำ

งานวิจัยฉบับนี้จัดทำขึ้นเพื่อศึกษารูปแบบการเชื่อมความสัมพันธ์ของข้อมูลที่จัดเก็บในฐานข้อมูลมอง
โคดีบีซึ่งเป็นฐานข้อมูลประเภทโนเอสคิวแอลที่ไม่รองรับการใช้คำสั่งเพื่อสอบถามข้อมูลที่มี
ความสัมพันธ์กันแต่ข้อมูลถูกแยกเก็บคนละส่วน โดยได้เสนอแนวทางการเชื่อมความสัมพันธ์ของ
ข้อมูลโดยการเขียนโปรแกรมร่วมกับการประยุกต์ใช้แมพรีดิวซ์เพื่อพัฒนาโปรแกรมเป็นไลบรารี
สำหรับเรียกใช้งานจากระบบของผู้ใช้งาน โดยมีการเปรียบเทียบการเชื่อมความสัมพันธ์ของข้อมูลจาก
รูปแบบที่นำเสนอกับรูปแบบการเชื่อมความสัมพันธ์ของข้อมูลจากการเขียนโปรแกรมโดยตรงภายใน
ระบบของผู้ใช้งาน

1.1 ความสำคัญและที่มาของงานวิจัย

ในโลกยุคปัจจุบันนี้เป็นยุคแห่งข้อมูลดิจิทัลซึ่งมีการใช้งานแลกเปลี่ยนข้อมูลข่าวสารระหว่างกันโดยมี
อัตราการใช้งานที่เพิ่มขึ้นและเติบโตอย่างรวดเร็วและได้กลายเป็นความซับซ้อนที่เพิ่มมากขึ้นใน
เรื่องของความหลากหลายของโครงสร้างการจัดเก็บข้อมูล ในการจัดเก็บข้อมูลนั้นมีทั้งแบบการ
จัดเก็บข้อมูลที่มีโครงสร้างแน่นอน การจัดเก็บข้อมูลที่ไม่มีโครงสร้างและการจัดเก็บข้อมูลที่มี
โครงสร้างแบบผสม ทำให้การสร้างความสัมพันธ์ของข้อมูลทำได้ค่อนข้างยาก หากมีการ
เปลี่ยนแปลงความต้องการก็อาจเกิดผลกระทบต่อความสัมพันธ์ของข้อมูลที่ได้สร้างไว้แล้ว ซึ่งความ
ต้องการขยายขนาดพื้นที่จัดเก็บข้อมูลที่มีปริมาณมากขึ้นเรื่อยๆ [1] ความต้องการในการรองรับ
ปริมาณผู้ใช้งานจำนวนมากพร้อมกัน ความต้องการในการเข้าถึงข้อมูลอย่างรวดเร็ว ความต้องการ
เหล่านี้ทำให้ฐานข้อมูลเชิงสัมพันธ์ทำได้จำกัดและหากดำเนินการได้ก็มีค่าใช้จ่ายที่ค่อนข้างสูง ดังนั้น
จึงมีฐานข้อมูลที่ไม่ใช้หลักการของฐานข้อมูลเชิงสัมพันธ์เกิดขึ้นมา เรียกว่าโนเอสคิวแอล (NoSQL)
เพื่อรองรับกับความต้องการที่กล่าวมาข้างต้น

แนวคิดของโนเอสคิวแอลถูกเสนอขึ้นมาในปี ค.ศ. 2009 เพื่อข้ามข้อจำกัดของฐานข้อมูลเชิงสัมพันธ์
[2] และเป็นตัวเลือกให้กับสถานการณ์ที่หลากหลายของระบบต่างๆในปัจจุบัน โดยเฉพาะในโลกของ
อินเทอร์เน็ตที่ความต้องการทางด้านการขยายขนาดพื้นที่จัดเก็บข้อมูลมากกว่าความต้องการด้าน
ความสัมพันธ์ของข้อมูล รวมถึงความต้องการที่จะเข้าถึงข้อมูลอย่างรวดเร็วแม้ว่าในบางครั้งข้อมูลที่
ได้อาจยังไม่สมบูรณ์หรือเป็นปัจจุบัน แต่ผู้ใช้งานก็พอใจมากกว่าที่จะต้องรอข้อมูลจนสมบูรณ์หรือ
เป็นปัจจุบัน เพราะว่าฐานข้อมูลแบบโนเอสคิวแอลนั้นมีอัตราการอ่านและเขียนข้อมูลที่เร็วมากเมื่อ
เทียบการกับฐานข้อมูลเชิงสัมพันธ์แต่ไม่รับประกันความสมบูรณ์ของข้อมูล ลักษณะหนึ่งของ

ฐานข้อมูล โนเอสคิวแอลที่ต่างจากฐานข้อมูลเชิงสัมพันธ์คือไม่มีคำสั่งสอบถามข้อมูลที่สามารถเชื่อมความสัมพันธ์ของข้อมูลที่แยกเก็บคนละส่วนได้

การออกแบบข้อมูลให้มีความสัมพันธ์กันเป็นสิ่งที่น่าสนใจในฐานข้อมูลเชิงสัมพันธ์เพื่อทำให้การบริหารจัดการข้อมูลมีประสิทธิภาพมากขึ้น ข้อมูลมีความถูกต้องสูง แต่การออกแบบฐานข้อมูลที่มีความสัมพันธ์กันมากจะทำให้การสอบถามข้อมูลช้าลงเพราะต้องทำการเชื่อมความสัมพันธ์ของข้อมูลที่มีความเกี่ยวข้องกัน ในขณะที่ภาษาที่ใช้ในการสอบถามข้อมูลของฐานข้อมูล โนเอสคิวแอลยังไม่รองรับการเชื่อมความสัมพันธ์ของข้อมูลที่สัมพันธ์กันแต่อยู่คนละส่วน ทำให้ผู้ที่ออกแบบระบบต้องออกแบบให้ข้อมูลที่เกี่ยวข้องกันอยู่ด้วยกันทั้งหมดเพื่อให้รองรับกับคำสั่งที่ใช้สอบถามข้อมูลเป็นผลให้การสอบถามข้อมูลทำได้รวดเร็วเนื่องจากไม่ต้องสูญเสียเวลาในเชื่อมความสัมพันธ์ แต่การออกแบบให้ข้อมูลอยู่ด้วยกันทั้งหมดนั้นก็ยังมีข้อด้อยตรงที่การปรับปรุงข้อมูลต้องใช้เวลาานมากกว่าปกติเพราะต้องปรับปรุงทุกรายการและหากต้องการข้อมูลเพียงบางส่วนของรายการข้อมูลนั้น ก็ต้องเข้าถึงข้อมูลรายการนั้นทั้งหมดทำให้สูญเสียเวลาในส่วนการเข้าถึงข้อมูลที่ไม่จำเป็น แต่ถ้ามีการออกแบบข้อมูลให้แยกเก็บคนละส่วนบนฐานข้อมูล โนเอสคิวแอล ผู้พัฒนาระบบจะต้องทำการเขียนโปรแกรมเพื่อเชื่อมความสัมพันธ์ของข้อมูลเหล่านั้นเองซึ่งหากไม่มีความเชี่ยวชาญอาจทำให้การทำงานของโปรแกรมเกิดข้อผิดพลาดได้ ทั้งนี้การเขียนโปรแกรมเพื่อเชื่อมความสัมพันธ์เองทำให้สูญเสียเวลาในการโอนถ่ายข้อมูลระหว่างระบบผู้ใช้งานกับฐานข้อมูล สูญเสียเวลาในกระบวนการเชื่อมความสัมพันธ์

งานวิจัยนี้จึงได้นำเสนอรูปแบบการสร้างความสัมพันธ์ของฐานข้อมูล โนเอสคิวแอลขึ้นมาเพื่อช่วยให้การสอบถามข้อมูลที่อยู่คนละส่วนทำได้สะดวกขึ้น การเขียนโปรแกรมทำได้ง่ายขึ้น ลดการสูญเสียเวลาในการโอนถ่ายข้อมูลมาที่ระบบเพราะกระบวนการทำงานอยู่ที่ฝั่งฐานข้อมูลเป็นหลัก ในงานวิจัยนี้ได้ใช้ฐานข้อมูลมองโกดีบีเป็นกรณีศึกษาเนื่องจากฐานข้อมูลมองโกดีบีสามารถเข้าถึงข้อมูลได้เร็วมาก ซึ่งในข้อมูลกว่า 50 กิกะไบต์ ฐานข้อมูลมองโกดีบีสามารถเข้าถึงข้อมูลได้เร็วกว่าฐานข้อมูลมายเอสคิวแอลถึง 10 เท่า [3] นอกจากนี้ฐานข้อมูลมองโกดีบียังมีหลายองค์ครที่เลือกใช้ [4] ซึ่งรูปแบบที่ได้เสนอในงานวิจัยเป็นการนำแมพรีดิวซ์ (MapReduce) มาช่วยเชื่อมความสัมพันธ์ของข้อมูล ซึ่งในการทดลองนั้นจะมีการออกแบบโครงสร้างข้อมูลให้มีความสัมพันธ์กันแล้วทำการสอบถามข้อมูลไปยังฐานข้อมูลเพื่อให้ได้ผลลัพธ์ตามที่กำหนด แล้วนำค่าที่ได้จากการทดลองนำมาเปรียบเทียบพร้อมทั้งอธิบาย วิเคราะห์คุณภาพของการประมวลผลที่ได้จากการทดลองในงานวิจัยนี้

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อศึกษารูปแบบการจัดการข้อมูล สถาปัตยกรรมของฐานข้อมูลแบบ โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์ (Document)
- 2) เพื่อศึกษาการเขียนโปรแกรมจัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์
- 3) เพื่อพัฒนารูปแบบการเชื่อมความสัมพันธ์ของข้อมูล ในฐานข้อมูล โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์
- 4) เพื่อศึกษาและเปรียบเทียบผลลัพธ์ที่ได้จากพัฒนารูปแบบการเชื่อมความสัมพันธ์ของข้อมูล โดยการใช้รูปแบบเชื่อมความสัมพันธ์ของข้อมูลด้วยแมพรีดิวซ์กับรูปแบบเชื่อมความสัมพันธ์ของข้อมูลด้วยการเขียนโปรแกรมโดยตรงภายในระบบผู้ใช้งาน

1.3 ประโยชน์และผลที่คาดว่าจะได้รับจากงานวิจัย

- 1) ทำให้เข้าใจรูปแบบการจัดเก็บข้อมูล สถาปัตยกรรมของฐานข้อมูล โนเอสคิวที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์
- 2) ทำให้เข้าใจวิธีการประมวลผลข้อมูลปริมาณมากด้วยแมพรีดิวซ์
- 3) รูปแบบการเชื่อมความสัมพันธ์ของข้อมูล ในฐานข้อมูล โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์
- 4) นำรูปแบบการเชื่อมความสัมพันธ์ของข้อมูลของฐานข้อมูล โนเอสคิวแอลไปประยุกต์ใช้

1.4 ขอบเขตงานวิจัย

- 1) ศึกษาารูปแบบการเชื่อมความสัมพันธ์ของข้อมูลจากการเขียนโปรแกรมโดยตรงภายในระบบของผู้ใช้งานของฐานข้อมูล โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์
- 2) ใช้แมพรีดิวซ์มาประยุกต์ใช้ในรวมผลลัพธ์จากการสร้างความสัมพันธ์ของฐานข้อมูล โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์บนเฟรมเวิร์คของฐานข้อมูลมองโกดีบี
- 3) ใช้ภาษาซีชาร์ป (C#) สำหรับสร้างไลบรารีที่มีกระบวนการเชื่อมความสัมพันธ์ของข้อมูลของฐานข้อมูล โนเอสคิวแอลที่มีการจัดเก็บข้อมูลแบบด็อคคิวเมนต์

1.5 ขั้นตอนการดำเนินงาน

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาทั้งหมดออกเป็น 5 บทด้วยกัน โดยแต่ละบทมีการนำเสนอเนื้อหาดังรายละเอียดต่อไปนี้

บทที่ 1 บรรยายความสำคัญและที่มาของการทำวิจัย ปัญหาที่ทำให้เกิดงานวิจัยนี้ วัตถุประสงค์ของงานวิจัย ประโยชน์และผลที่คาดว่าจะได้รับจากการทำงานวิจัย และขอบเขตในการทำงานวิจัยนี้

บทที่ 2 อธิบายทฤษฎีพื้นฐานที่เกี่ยวข้อง ซึ่งประกอบด้วยหัวข้อที่เกี่ยวข้องกับฐานข้อมูลโนเอสคิวแอล ฐานข้อมูลมองโกดีบี การเขียนโปรแกรมจัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์ ตลอดจนกล่าวถึงงานวิจัยที่เกี่ยวข้อง

บทที่ 3 อธิบายหลักการพัฒนารูปแบบการเชื่อมความสัมพันธ์ของข้อมูลของฐานข้อมูลโนเอสคิวแอลประเภทคอกิวเมนต์โดยนำแมพรีดิวซ์มาช่วยในการรวมผลลัพธ์จากการสอบถามข้อมูลที่มีความสัมพันธ์กันเพื่อลดระยะเวลาที่ต้องโอนถ่ายข้อมูลมาเชื่อมความสัมพันธ์ที่ระบบ โดยกล่าวถึงขั้นตอนการทำงานอย่างละเอียดในงานวิจัยนี้

บทที่ 4 แสดงรูปแบบการทดลองที่ใช้ทดสอบวัดประสิทธิภาพของเวลาที่ใช้ในการประมวลผลของรูปแบบการเชื่อมความสัมพันธ์ของข้อมูลที่นำเสนอในงานวิจัยนี้เปรียบเทียบกับรูปแบบการเชื่อมความสัมพันธ์ของข้อมูลจากการเขียนโปรแกรมภายในระบบโดยตรง แล้วแสดงผลที่ได้จากการทดลองพร้อมทั้งอธิบายและวิเคราะห์ผลจากค่าเวลาที่ใช้ในการประมวลผล

บทที่ 5 สรุปผลที่ได้รับจากการทำวิจัยนี้ และกล่าวถึงงานวิจัยนี้เป็นแนวทางพัฒนางานสืบต่อไปในอนาคต

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในเนื้อหาของบทนี้กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ ผู้วิจัยได้อธิบายความหมายและความสำคัญของเนื้อหาที่เกี่ยวข้องในการศึกษาค้นคว้าวิจัย เพื่อนำความรู้ที่ได้ดำเนินการศึกษานั้นมาใช้อ้างอิงในการทำงานวิจัยในครั้งนี้ โดยได้อธิบายเนื้อหาทฤษฎี และงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 ความรู้พื้นฐานที่เกี่ยวข้อง

ในงานวิจัยนี้ได้นำเสนอการเชื่อมความสัมพันธ์ของคอลเลกชันของฐานข้อมูลมอดโกดิบี โดยได้อธิบายความรู้พื้นฐานที่เกี่ยวข้องกับงานวิจัยที่นำเสนอเพื่อให้ผู้ศึกษาสามารถเข้าใจกระบวนการในงานวิจัยนี้ได้ดียิ่งขึ้น ซึ่งความรู้พื้นฐานที่เกี่ยวข้องนั้นประกอบไปด้วย ฐานข้อมูลโนเอสคิวแอล ฐานข้อมูลมอดโกดิบี อัลกอริทึมแมพรีดิวซ์ ซึ่งจะได้อธิบายโดยมีรายละเอียดดังนี้

2.1.1 ฐานข้อมูลโนเอสคิวแอล (NoSQL Database)

โนเอสคิวแอลมาจากการรวมคำศัพท์สองคำคือ No และ SQL ผู้คิดค้นหรือผู้ที่นำคำศัพท์ NoSQL มาใช้ตอนแรกต้องการที่จะใช้คำว่า No RDBMS หรือ No relational แต่ด้วยการออกเสียงที่ดูน่าฟังกว่าจึงได้ใช้คำว่า NoSQL ซึ่งมีบางคนพยายามนำเสนอว่า NoSQL นั้นย่อมาจาก Not Only SQL ซึ่งไม่ว่าจะหมายถึงอย่างไรก็ตาม NoSQL ก็ถูกใช้แทนฐานข้อมูลและการจัดเก็บข้อมูลที่ไม่เหมือนกับหลักการของฐานข้อมูลเชิงสัมพันธ์ [5] สำหรับการออกแบบการจัดเก็บข้อมูลในฐานข้อมูลโนเอสคิวแอลนั้นไม่ต้องมีกระบวนการลดความซ้ำซ้อนของข้อมูล (Normalize) เพราะเป็นฐานข้อมูลที่สามารถเก็บข้อมูลที่มีความสัมพันธ์กันไว้ภายในระเบียบเดียวกัน ไม่มีการกำหนดโครงสร้างการจัดเก็บข้อมูลที่เฉพาะเจาะจง

ในปีคริสต์ศักราช 1998 Carlo Strozzi ได้คิดค้นฐานข้อมูลเชิงสัมพันธ์โอเพนซอร์สที่มีการทำงานแบบเบา (Light Weight) โดยออกแบบให้ฐานข้อมูลเก็บตารางเป็นแบบแฟ้มแอสกี (ASCII File) ข้อมูลแต่ละระเบียบจะเป็นการเก็บแบบบรรทัดที่มีการแบ่งฟิลด์ข้อมูลด้วยแถบช่องว่าง (Tab) ในการค้นหาหรือกระทำการใดๆกับข้อมูลก็จะไม่ใช้ภาษาเอสคิวแอลแต่จะใช้คำสั่ง Shell Script ในการดำเนินการ จึงเป็นที่มาของคำว่า “โนเอสคิวแอล” [6] เป้าหมายของฐานข้อมูลโนเอสคิวแอลคือลดความยุ่งยากซับซ้อนในการจัดการข้อมูลเพื่อตอบสนองการทำงานได้อย่างรวดเร็วขึ้น รองรับการขยายขนาดของข้อมูล ในปัจจุบันฐานข้อมูลโนเอสคิวแอลเข้ามามีบทบาทมากในการจัดเก็บและจัดการข้อมูลที่มีปริมาณมาก มีหลายองค์กรที่นำฐานข้อมูลโนเอสคิวแอลมาใช้งาน เช่น กูเกิล (Google) ทวิตเตอร์

(Twitter) เฟสบุ๊ก (Facebook) เป็นต้น เพราะแอปพลิเคชันขององค์กรเหล่านี้มีจำนวนผู้ใช้งานมหาศาล จำนวนข้อมูลที่ต้องจัดเก็บก็มากตามไปด้วย ดังนั้นองค์กรเหล่านี้จึงต้องการฐานข้อมูลที่รองรับการจัดเก็บข้อมูลที่มีปริมาณมาก เน้นประสิทธิภาพการจัดเก็บและการค้นหาข้อมูลที่รวดเร็วและต้องสามารถทำซ้ำเพื่อสำรองข้อมูลที่มีปริมาณมากไปยังเครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ

ฐานข้อมูลโนเอสคิวแอลสามารถจำแนกออกเป็น 4 แบบได้ดังต่อไปนี้ [7]

- 1) Key-Value Data Model
- 2) Document Data Model
- 3) Column-Family Data Model
- 4) Graph Data Model

แบบที่ 1 Key-Value Data Model

ฐานข้อมูลโนเอสคิวแอลแบบ Key-Value [7] เป็นรูปแบบที่มีการนำคีย์ (Key) หรือ ID (Identity) เข้ามาใช้ในการจัดเก็บ การระบุระเบียบข้อมูลและการดึงข้อมูลออกมาใช้งาน ในแต่ละคอลัมน์จะเก็บค่าคีย์คู่กับค่าของข้อมูลเสมอ ตัวอย่างการจัดเก็บข้อมูลแบบ Key-Value ดังรูปที่ 2.1

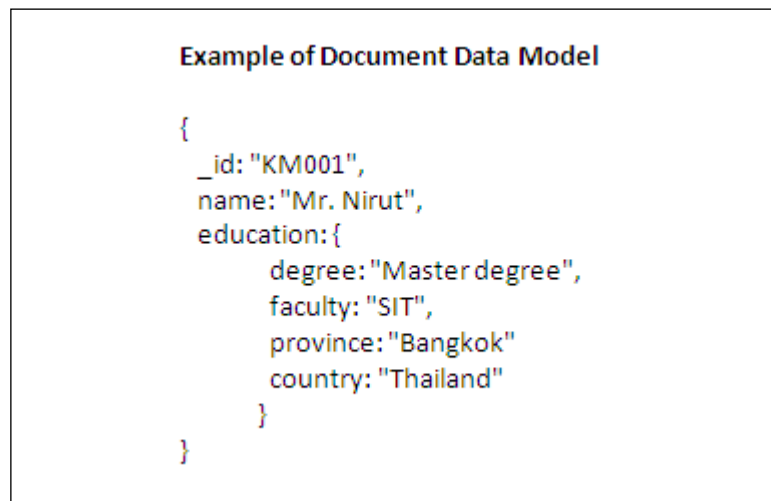
Example of Key-Value Data Model of NoSQL		
Key : 1	ID : KM001	Name : Mr. Nirut
Key : 2	Degree : Master degree	Faculty : SIT
Key : 3	Province : Bangkok	Country : Thailand

รูปที่ 2.1 ตัวอย่างการจัดเก็บข้อมูลแบบ Key-Value Data Model

จากรูปที่ 2.1 แสดงการจัดเก็บข้อมูลแบบ Key-Value สังเกตได้ว่าในหนึ่งระเบียบของแต่ละคอลัมน์จะมีการเก็บค่าคีย์คู่กับค่าข้อมูลเสมอ ฐานข้อมูลประเภทโนเอสคิวแอลที่มีลักษณะการจัดการข้อมูลแบบ Key-Value ได้แก่ Redis [20], Riak [21] และ Tokyo Cabinet [22] เป็นต้น

แบบที่ 2 Document Data Model

ฐานข้อมูลโนเอสคิวแอลแบบ Document Data Model เป็นรูปแบบที่มีการจัดเก็บข้อมูลในลักษณะแบบคอกิวเมนต์ที่มีโครงสร้างทางมิติเชิงลึกและเชิงกว้างในการจัดเก็บข้อมูล ซึ่งการเก็บข้อมูลนั้นสามารถเก็บในรูปแบบ XML (Extensible Markup Language), JSON (Java Script Object Notation), BSON (Binary JSON) การจัดเก็บข้อมูลในฐานข้อมูลนั้นใช้หลักการใส่คีย์ในคอกิวเมนต์ เพื่อเป็นการอ้างอิงข้อมูล โครงสร้างการจัดเก็บข้อมูลเป็นแบบลำดับชั้นซึ่งสามารถเก็บข้อมูลเชิงเดี่ยวและข้อมูลเชิงลึกได้ สามารถเก็บค่าที่เป็นอาร์เรย์ (Array) ได้ ฐานข้อมูลโนเอสคิวแอลที่มีลักษณะการจัดการข้อมูลแบบ Document Data Model ได้แก่ MongoDB [23] และ CouchDB [24] เป็นต้น ตัวอย่างการจัดเก็บข้อมูลแบบ Document Data Model ดังรูปที่ 2.2



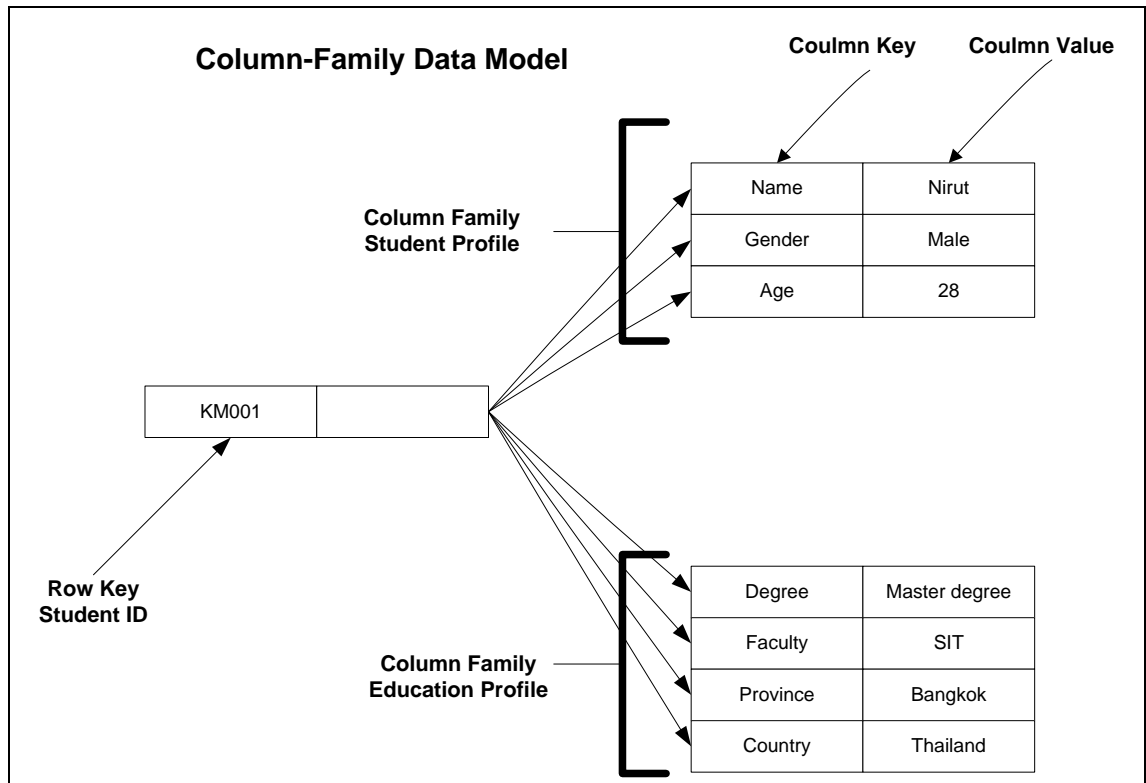
รูปที่ 2.2 ตัวอย่างการจัดเก็บข้อมูลแบบ Document Data Model

จากรูปที่ 2.2 แสดงการจัดเก็บข้อมูลแบบ Document Data Model ของฐานข้อมูลโนเอสคิวแอลซึ่งจะเห็นได้ว่าการจัดเก็บนั้นเป็นลักษณะการจัดเก็บข้อมูลแบบโครงสร้างของข้อมูลหนึ่งระเบียน มีคีย์กำกับค่าของข้อมูล ข้อมูลในแต่ละฟิลด์สามารถเก็บเป็นข้อมูลเชิงเดี่ยวได้เช่น ฟิลด์ Name เก็บข้อมูลตัวอักษร “Mr. Nirut” บางฟิลด์ก็สามารถเก็บข้อมูลเชิงลึกได้เช่น ฟิลด์ Education สามารถเก็บข้อมูล Degree, Faculty, Province และ Country เป็นชั้นข้อมูลซ้อนลงไปได้ เป็นต้น

แบบที่ 3 Column-Family Data Model

ฐานข้อมูลโนเอสคิวแอลแบบ Column-Family Data Model เป็นรูปแบบที่มีการจัดเก็บข้อมูลในลักษณะแบบตารางที่มีการจัดกลุ่มของคอลัมน์ โดยแต่ละตารางสามารถประกอบด้วยหลายแถวและแต่ละแถวสามารถประกอบด้วยจำนวนคอลัมน์ที่แตกต่างกันได้ ไม่จำเป็นว่าทุกแถวจะมีจำนวน

คอลัมน์คงที่เท่ากันตลอดทุกแถว ในแต่ละแถวนี้จะมีคีย์กำกับ การดำเนินการจัดเก็บข้อมูลจะมีการจัดเก็บข้อมูลแบบเรียงลำดับ โดยจะใช้คีย์ของแถวข้อมูลเป็นตัวจัดเรียงค่าข้อมูล ฐานข้อมูล โนเอสคิวแอลที่มีลักษณะการจัดการข้อมูลแบบ Column-Family Data Model ได้แก่ Cassandra [25] และ HBase [26] เป็นต้น ตัวอย่างการจัดเก็บข้อมูลแบบ Column-Family Data Model ดังรูปที่ 2.3



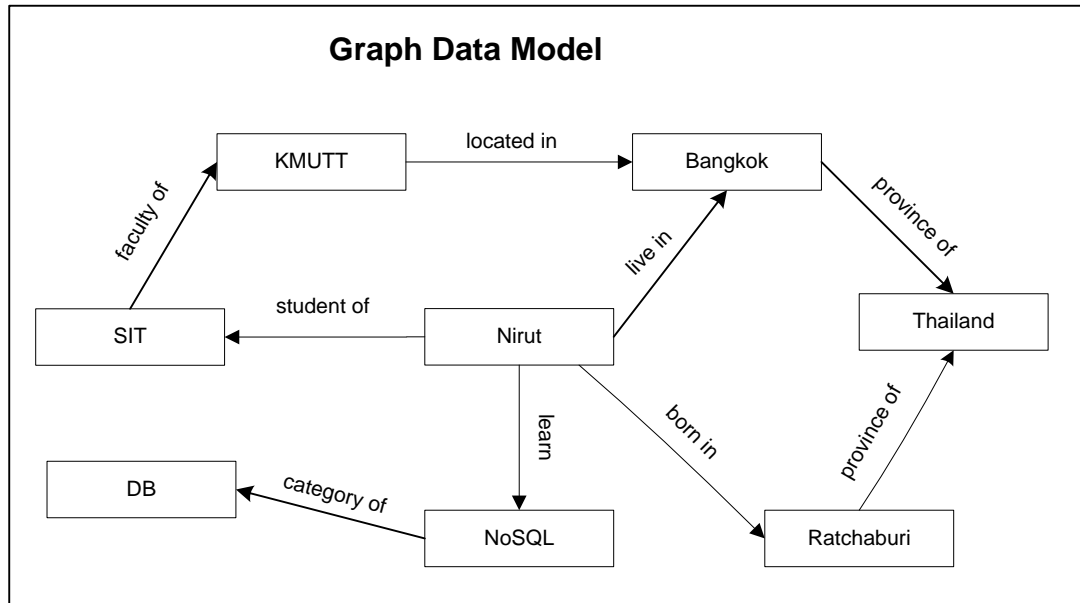
รูปที่ 2.3 ตัวอย่างการจัดเก็บข้อมูลแบบ Column-Family Data Model

จากรูปที่ 2.3 แสดงถึงการจัดเก็บข้อมูลแบบ Column-Family Data Model ในตารางข้อมูล ประกอบด้วยแถวข้อมูลที่มีคีย์กำกับ ในที่นี้ข้อมูลนักเรียนมีคีย์ของแถวข้อมูลคือ KM001 แต่ละแถวจะมีคีย์ที่เชื่อมไปยังกลุ่มของคอลัมน์ ในที่นี้มีกลุ่มคอลัมน์อยู่สองกลุ่มคือกลุ่มคอลัมน์ประวัติส่วนตัวนักเรียนและกลุ่มคอลัมน์ประวัติการศึกษา ซึ่งแต่ละกลุ่มของคอลัมน์จะประกอบด้วยฟิลด์ของข้อมูล แต่ละฟิลด์ของข้อมูลประกอบไปด้วยคีย์ของคอลัมน์และค่าข้อมูลในคอลัมน์นั้น

แบบที่ 4 Graph Data Model

ฐานข้อมูล โนเอสคิวแอลแบบ Graph Data Model เป็นรูปแบบที่มีการจัดเก็บข้อมูลที่มีความสัมพันธ์ในลักษณะแบบกราฟซึ่งมีลักษณะการเชื่อมโยงโหนด (Node) ด้วยจุดเชื่อมต่อ (Edge) เช่น ข้อมูลประเภทเส้นทางแผนที่ ข้อมูลการเดินทางสาธารณะ เป็นต้น การจัดเก็บข้อมูลชนิดนี้จะเป็นการเก็บค่า

ข้อมูลโหนดและค่าข้อมูลจุดเชื่อมต่อ การจัดเก็บข้อมูลแบบ Graph Data Model สนับสนุนการจัดเก็บค่าที่มีความสัมพันธ์แบบ Single-Depth การค้นหาข้อมูลสามารถท่องโหนดลงไปได้เรื่อยๆตามความสัมพันธ์ของข้อมูล ฐานข้อมูล โนเอสคิวแอลที่มีลักษณะการจัดการข้อมูลแบบ Graph Data Model ได้แก่ Neo4J [27] และ FlockDB [28] เป็นต้น ตัวอย่างการจัดเก็บข้อมูลแบบ Graph Data Model ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างการจัดเก็บข้อมูลแบบ Graph Data Model

จากรูปที่ 2.4 แสดงถึงข้อมูลแบบ Graph Data Model เป็นตัวอย่างความสัมพันธ์ของข้อมูลในรูปแบบที่เป็นการเชื่อมโยงแบบกราฟข้อมูลแต่ละส่วนล้วนเกี่ยวข้องกัน ซึ่งฐานข้อมูล โนเอสคิวแอลสามารถรองรับการจัดเก็บข้อมูลที่มีความสัมพันธ์แบบนี้ได้

ในงานวิจัยนี้ได้เลือกใช้ฐานข้อมูลมอโกดีบีเพราะเป็นฐานข้อมูลที่ได้รับความนิยมค่อนข้างมาก [8] และมีลักษณะการจัดเก็บข้อมูลแบบคอกคิวเมนต์ที่มีความใกล้เคียงกับฐานข้อมูลเชิงสัมพันธ์มากกว่ารูปแบบอื่น ดังนั้นการเลือกฐานข้อมูล โนเอสคิวแอลประเภทนี้ทำให้ผู้ที่ต้องการเปลี่ยนการใช้งานฐานข้อมูลเชิงสัมพันธ์มาใช้ฐานข้อมูล โนเอสคิวแอลประเภทนี้สามารถเรียนรู้การใช้งานได้เร็วขึ้น

2.1.2 มองโกดีบี (MongoDB)

ฐานข้อมูลมองโกดีบีเป็นฐานข้อมูลจัดอยู่ในกลุ่ม Document Data Model ของฐานข้อมูลโนเอสคิวแอล ซึ่งฐานข้อมูลมองโกดีบีถูกพัฒนาจากองค์กร 10gen เป็นฐานข้อมูลแบบโอเพนซอร์ส (Open Source) พัฒนาการจัดการข้อมูลด้วยภาษาซี [7] โดยผู้ที่สนใจสามารถดาวน์โหลดโปรแกรมฐานข้อมูลมองโกดีบีไปใช้งานได้ ฐานข้อมูลมองโกดีบีมีคุณสมบัติในการจัดการข้อมูลดังนี้

- ฐานข้อมูลมองโกดีบีเป็นฐานข้อมูลที่มีรูปแบบการจัดการข้อมูลที่เรียนรู้ได้ง่ายเมื่อเปรียบเทียบกับฐานข้อมูลโนเอสคิวแอลในรูปแบบอื่นๆ เช่น Key-Value Data Model, Column-Family Data Model และ Graph Data Model ฐานข้อมูลมองโกดีบียังมีหลักการแนวคิดจัดการข้อมูลพื้นฐานส่วนใหญ่คล้ายกับฐานข้อมูลเชิงสัมพันธ์ ทำให้ผู้ใช้งานที่มีพื้นฐานจากการใช้งานฐานข้อมูลเชิงสัมพันธ์มาก่อนสามารถทำความเข้าใจหลักการของฐานข้อมูลมองโกดีบีได้ไม่ยาก
- ผู้ที่นำฐานข้อมูลมองโกดีบีไปใช้งานนั้นสามารถปรับเข้ากับลักษณะการทำงานได้ง่าย เพราะฐานข้อมูลมองโกดีบีมีหลักการจัดการโครงสร้างข้อมูลที่ยืดหยุ่น ผู้ใช้งานไม่ต้องประกาศโครงสร้างในการจัดเก็บข้อมูลก่อนนำค่าข้อมูลจัดเก็บลงในฐานข้อมูลซึ่งเป็นวิธีการที่เหมาะสมกับการจัดเก็บข้อมูลแบบไม่ต้องมีโครงสร้าง ลดความซับซ้อนในการจัดการข้อมูลที่มีปริมาณมาก
- ในส่วนของการรองรับเพื่อขยายขนาดพื้นที่จัดเก็บข้อมูล ฐานข้อมูลมองโกดีบีมีคุณสมบัติในการจัดการข้อมูลจำนวนมากได้อย่างดี ในขณะที่ขนาดและเส้นทางการรับส่งข้อมูลยังคงเติบโตขึ้นเรื่อยๆ แต่กลับไม่ค่อยมีผลกระทบต่อระบบผู้ใช้งาน

ฐานข้อมูลมองโกดีบีมีหลักการทำงานในการจัดการข้อมูลซึ่งโดยหลักๆจะแบ่งการจัดเก็บข้อมูลออกเป็น 3 ส่วนดังต่อไปนี้

1) ฐานข้อมูล (Database) เป็นการจัดเก็บข้อมูลเป็นกลุ่มโดยในฐานข้อมูลประกอบด้วยกลุ่มข้อมูลคอลเลกชัน เป็นการเก็บข้อมูลคอลเลกชันเดียวหรือรวบรวมหลายๆคอลเลกชันเก็บไว้ในฐานข้อมูลได้

2) คอลเลกชัน (Collection) เป็นการจัดเก็บข้อมูลเป็นกลุ่มซึ่งอยู่ภายใต้ฐานข้อมูล โดยในคอลเลกชันประกอบด้วยกลุ่มข้อมูลคอกิวเมนต์ ซึ่งหลักการจัดเก็บของคอลเลกชันจะคล้ายกับการจัดเก็บแบบตารางในฐานข้อมูลเชิงสัมพันธ์ การจัดเก็บมีส่วนที่ไม่เหมือนกับตารางในฐานข้อมูลเชิงสัมพันธ์ ตรงที่คอลเลกชันไม่จำเป็นต้องสร้างโครงสร้างการจัดเก็บข้อมูลที่แน่นอนก่อนนำค่าของข้อมูลเก็บไว้ในฐานข้อมูล

3) คอกิวเมนต์ (Document) เป็นการจัดเก็บข้อมูลซึ่งอยู่ภายในคอลเลกชัน โดยในคอกิวเมนต์ประกอบไปด้วยค่าของข้อมูล ที่มีลักษณะเป็น Key-Value pairs โดย Key เป็นค่าเพื่อใช้อ้างอิง

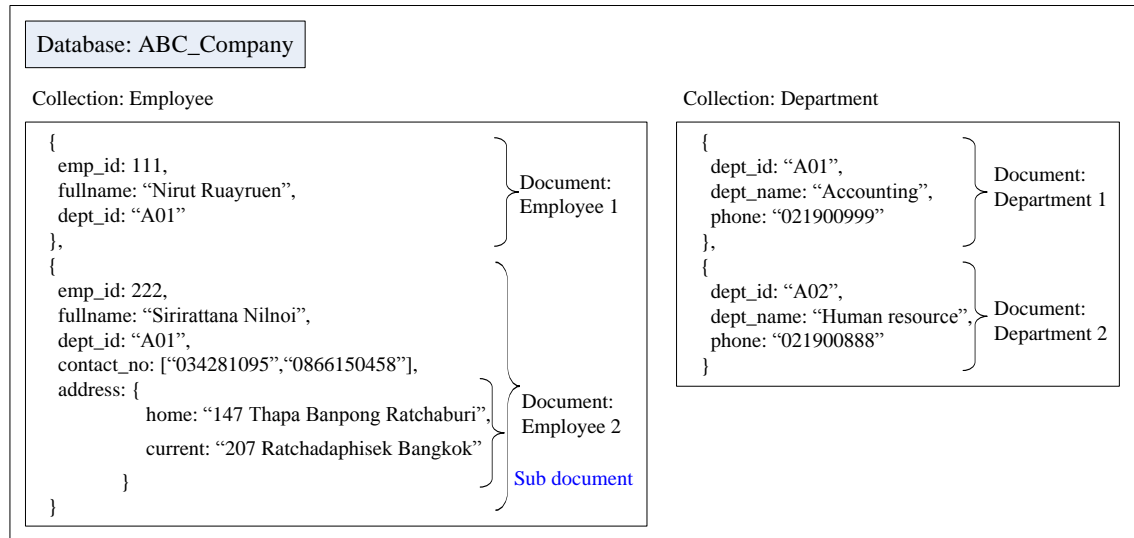
และ Value เป็นค่าของข้อมูล ซึ่งสามารถเป็นประเภทของข้อมูลต่างๆ ได้เช่น ตัวอักษร (String) ตัวเลขจำนวนเต็ม (Integer) ตัวเลขทศนิยม (Float) อาร์เรย์ (Array) เป็นต้น อีกทั้งในค่า Value ยังสามารถเก็บค่าข้อมูลที่เป็นแบบดอควิวเมนต์ซ้อนลงไปได้อีกด้วย

จากคู่มือฐานข้อมูลมองโกดีบีขององค์กรมองโกดีบี [9] ได้อธิบายเนื้อหาหลักการจัดเก็บข้อมูลของฐานข้อมูลมองโกดีบีเปรียบเทียบกับหลักการเก็บข้อมูลของฐานข้อมูลเชิงสัมพันธ์ได้ดังรูปที่ 2.5

Terminology and Concepts	
The following table presents the various SQL terminology and concepts and the corresponding MongoDB terminology and concepts.	
SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON document
column	field
index	index
table joins	embedded documents and linking
primary key	primary key
Specify any unique column or column combination as primary key.	In MongoDB, the primary key is automatically set to the <code>_id</code> field.

รูปที่ 2.5 หลักการเก็บข้อมูลของฐานข้อมูลมองโกดีบีกับข้อมูลของฐานข้อมูลเชิงสัมพันธ์ [9]

ในรูปที่ 2.5 แสดงการเปรียบเทียบหลักการเก็บข้อมูลของฐานข้อมูลเชิงสัมพันธ์กับฐานข้อมูลมองโกดีบีโดยมีส่วนที่เรียกชื่อแตกต่างกัน เช่น Table ในฐานข้อมูลเชิงสัมพันธ์เทียบได้กับ Collection ในฐานข้อมูลมองโกดีบี Row ในฐานข้อมูลเชิงสัมพันธ์เทียบได้กับ Document ในฐานข้อมูลมองโกดีบี Column ในฐานข้อมูลเชิงสัมพันธ์เทียบได้กับ Field ในฐานข้อมูลมองโกดีบี table joins ในฐานข้อมูลเชิงสัมพันธ์เทียบได้กับการเก็บข้อมูลแบบซ้อนกันของ Documents ในฐานข้อมูลมองโกดีบี เป็นต้น



รูปที่ 2.6 ตัวอย่างการจัดเก็บข้อมูลของฐานข้อมูลมองโกดีบี

จากรูปที่ 2.6 สามารถอธิบายการจัดเก็บข้อมูลของฐานข้อมูลมองโกดีบีได้ว่า มีการสร้างฐานข้อมูลชื่อว่า ABC_Company ภายในฐานข้อมูลประกอบไปด้วยสองคอลเลกชัน คือคอลเลกชัน Employee และคอลเลกชัน Department โดยแต่ละคอลเลกชันมีรายละเอียดดังนี้ คอลเลกชัน Department ประกอบด้วยสองคอคิวเมนต์ คือคอคิวเมนต์ Department 1 และคอคิวเมนต์ Department 2 ซึ่งทั้งสองคอคิวเมนต์ประกอบด้วยชื่อฟิลด์ดังนี้ ฟิลด์ dept_id เก็บรหัสหน่วยงาน ฟิลด์ dept_name เก็บชื่อหน่วยงานและฟิลด์ phone เก็บเบอร์โทรศัพท์ ซึ่งทุกฟิลด์เป็นข้อมูลแบบตัวอักษร คอลเลกชัน Employee ประกอบด้วยสองคอคิวเมนต์ โดยคอคิวเมนต์แรกชื่อว่า Employee 1 ประกอบด้วยชื่อฟิลด์ดังนี้ ฟิลด์ emp_id เก็บรหัสพนักงานซึ่งเป็นข้อมูลแบบตัวเลข ฟิลด์ fullname เก็บชื่อพนักงานซึ่งเป็นข้อมูลแบบตัวอักษร ฟิลด์ dept_id เก็บข้อมูลรหัสหน่วยงานซึ่งเป็นข้อมูลแบบตัวอักษร ส่วนคอคิวเมนต์ที่สองชื่อว่า Employee 2 มีชื่อฟิลด์ที่เพิ่มขึ้นมาจากคอคิวเมนต์ Employee 1 ดังนี้ ฟิลด์ contact_no เก็บข้อมูลหมายเลขติดต่อซึ่งเป็นข้อมูลแบบอาร์เรย์สามารถเก็บข้อมูลได้หลายค่า และฟิลด์ address เก็บข้อมูลที่อยู่ซึ่งเป็นข้อมูลแบบคอคิวเมนต์จะเห็นได้ว่าฟิลด์นี้สามารถเก็บข้อมูลในรูปแบบคอคิวเมนต์ที่ซ้อนลงในฟิลด์ได้ จากตัวอย่างจะเห็นได้ว่าฐานข้อมูลมองโกดีบีมีความยืดหยุ่นทางโครงสร้างในการจัดเก็บข้อมูลค่อนข้างสูง

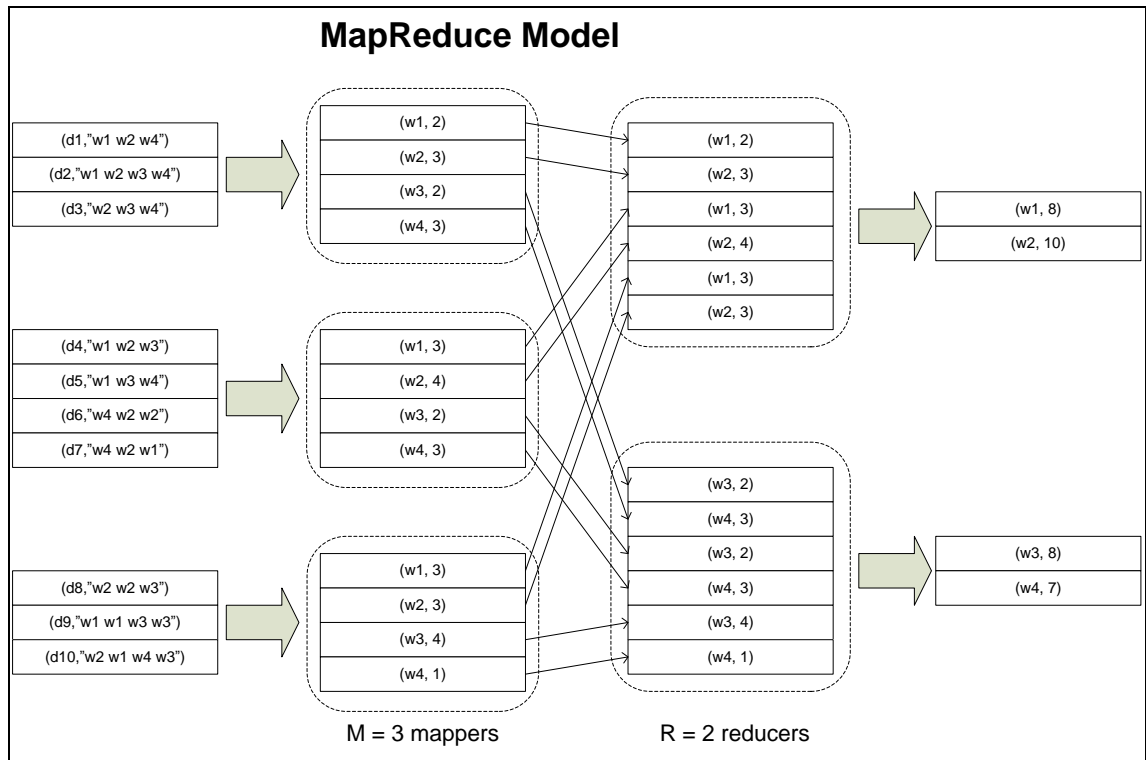
ด้วยรูปแบบการจัดเก็บข้อมูลตามที่แสดงในรูปที่ 2.6 นั้น ถ้าต้องการทราบว่าพนักงานแต่ละคนอยู่หน่วยงานใด ซึ่งฐานข้อมูลมองโกดีบียังไม่มีคำสั่งในการสอบถามข้อมูลที่มีความสัมพันธ์กันแต่เก็บอยู่คนละคอลเลกชัน หากต้องการข้อมูลดังกล่าวต้องอาศัยการเขียนโปรแกรมภายในระบบเอง ดังนั้น

งานวิจัยนี้จึงได้นำเสนอการนำหลักเขียน โปรแกรมแบบแมพรีดิวซ์มาประยุกต์ใช้ในการเชื่อม ความสัมพันธ์ของข้อมูลที่แยกเก็บคนละคอลเลกชัน

2.1.3 แมพรีดิวซ์ (MapReduce)

รูปแบบการเขียน โปรแกรมตามแนวคิดแมพรีดิวซ์ถูกพัฒนาขึ้น โดยของค์กรกูเกิล (Google) เพื่อเข้ามา ช่วยในการค้นหาข้อมูลที่มีปริมาณมาก และเข้ามาช่วยในการประมวลผลข้อความบนที่เก็บข้อมูลเว็บ ขนาดใหญ่ซึ่งข้อมูลเว็บเหล่านี้ถูกเก็บใน BigTable และระบบการเก็บไฟล์แบบกระจาย แนวคิดการ เขียนโปรแกรมแบบแมพรีดิวซ์ถูกออกแบบเพื่อการประมวลผลและสร้างข้อมูลที่มีขนาดใหญ่ด้วย วิธีการคำนวณแบบขนานพร้อมกันของหน่วยประมวลผล (Processor) จำนวนหนึ่งพันหน่วยในเวลา เดียวกัน [10]

การเขียนโปรแกรมตามแนวคิดแมพรีดิวซ์ที่เสนอ โดยกูเกิลนั้น กูเกิลได้เสนอตัวอย่างการนับจำนวน คำศัพท์ของข้อมูลจำนวนมากด้วยแมพรีดิวซ์ซึ่งเป็นการเขียน โปรแกรมที่มีทำงานแบบคู่ขนาน ตอบสนองความต้องการจัดการข้อมูลที่มีขนาดใหญ่ ตัวอย่างการใช้หลักการแมพรีดิวซ์ในการนับ จำนวนคำศัพท์ของข้อมูลจำนวนมาก [10] ได้อธิบายรายละเอียดดังรูปที่ 2.7 วัตถุประสงค์เพื่อนับ คำศัพท์ที่ซ้ำกันในที่เก็บข้อมูลที่มีปริมาณขนาดใหญ่ แมพรีดิวซ์มีกระบวนการทำงานแบ่งเป็นสอง ส่วน โดยส่วนแรกเป็นการทำงานของฟังก์ชันแมพ ซึ่งเป็นการกระจายการแมพข้อมูล ส่วนที่สองเป็น การทำงานของฟังก์ชันรีดิวซ์ ซึ่งเป็นการกระจายการลดจำนวนผลลัพธ์ของข้อมูล ในแต่ละส่วนจะมี การตั้งค่า M (Mapper Worker) เป็นค่าจำนวนหน่วยประมวลผลในการแมพข้อมูล และตั้งค่า R (Reducer Worker) เป็นค่าจำนวนหน่วยประมวลผลในการลดจำนวนผลลัพธ์ของข้อมูล ซึ่งมี Master Controller เป็นผู้ควบคุมกระจายงานให้ Mapper Worker และ Reducer Worker ในตัวอย่างโปรแกรม รายละเอียดการทำงานดังรูปที่ 2.7



รูปที่ 2.7 MapReduce Model Example Word Counting [10]

จากรูปที่ 2.7 สามารถอธิบายได้ว่า Master Controller มีการกำหนดให้ค่า M มีค่าเท่ากับ 3 คือแบ่ง Mapper Worker เท่ากับจำนวน 3 หน่วยประมวลผลและค่า R มีค่าเท่ากับ 2 คือแบ่ง Reducer Worker เท่ากับจำนวน 2 หน่วยประมวลผล ในส่วนการทำงานของฟังก์ชันแมพจะมีการแบ่งข้อมูลเป็นแต่ละเอกสาร แบ่งโดยใช้สูตร $1/M$ ของจำนวนข้อมูลที่ได้รับเข้ามาเพื่อส่งเอกสารให้ Mapper Worker แต่ละตัวอ่านค่าประมวลผล ในตัว Mapper Worker แต่ละตัวจะมีฟังก์ชันแมพในการคำนวณความถี่ของคำศัพท์ที่มีในแต่ละเอกสารที่ Mapper Worker อ่านและจะมีการบันทึกค่าความถี่ของคำศัพท์โดยบันทึกค่าแบบจัดเรียงค่าตามคำศัพท์ลงบนไฟล์ของระบบของ Mapper Worker แต่ละตัว โดยหลักการบันทึกไฟล์คือ Mapper Worker แต่ละตัวทำการบันทึกค่าคำศัพท์ตามจำนวนของ Reducer Worker ได้รับมอบหมายให้ทำการลดจำนวนผลลัพธ์ข้อมูลลงในระบบไฟล์โดยบันทึกหนึ่งไฟล์ต่อหนึ่ง Reducer Worker เมื่อ Mapper Worker บันทึกค่าเสร็จก็จะส่งคำร้องพร้อมข้อมูลที่อยู่ของไฟล์ที่บันทึกไปยัง Master Controller เพื่อให้ Master Controller ส่งงานให้กับส่วนการทำงานของฟังก์ชัน Reduce จะมีการมอบหมายให้ Reducer Worker แต่ละตัวทำการลดจำนวนผลลัพธ์ของข้อมูล โดย Reducer Worker ตัวแรกถูกสั่งให้จัดการกับข้อมูลคำศัพท์ W1 และ W2 ส่วน Reducer ตัวที่สองถูกสั่งให้จัดการกับข้อมูลคำศัพท์ W3 และ W4 การทำงานของ Reducer Worker แต่ละตัวเมื่อได้รับคำร้องจาก Master Controller เพื่อดึงข้อมูลคำศัพท์ที่ได้รับมอบหมายให้ลดจำนวนผลลัพธ์ของข้อมูลโดยการหาค่าผลรวมค่าความถี่ของคำศัพท์แต่ละคำ เมื่อ Reducer Worker ได้ค่าผลรวมเรียบร้อยแล้วก็จะบันทึกค่า

ลงในระบบไฟล์แล้ว Reducer Worker แต่ละตัวก็จะส่งผลไปยัง Master Controller เพื่อแสดงค่าผลลัพธ์

จากตัวอย่างการทำงานของโปรแกรมที่ใช้หลักการแมพรีดิวซ์จะเห็นได้ว่า ฟังก์ชันแมพ ประกอบไปด้วยการเตรียมข้อมูลเพื่อส่งต่อของหนึ่งเซตคู่ Key-Value ไปยังคู่อื่น ดังอัลกอริทึมด้านล่างนี้

$$\text{Map: } (k_1, v_1) \rightarrow [(k_2, v_2)]$$

ฟังก์ชันรีดิวซ์ เป็นการจัดกลุ่มข้อมูลผลลัพธ์ที่ได้จากฟังก์ชันแมพ หรือเรียกได้ว่าการลดผลลัพธ์ของฟังก์ชันแมพ โดยใช้คีย์ k_2 นำกลุ่มค่าข้อมูลของแต่ละคีย์ไปเข้าฟังก์ชัน f โดยคีย์ที่มีค่าตัวเดียวกันก็จะถูกลดและรวมค่าเข้าด้วยกัน ดังอัลกอริทึมด้านล่างนี้

$$\text{Reduce: } (k_2, [v_2]) \rightarrow (k_2, f([v_2]))$$

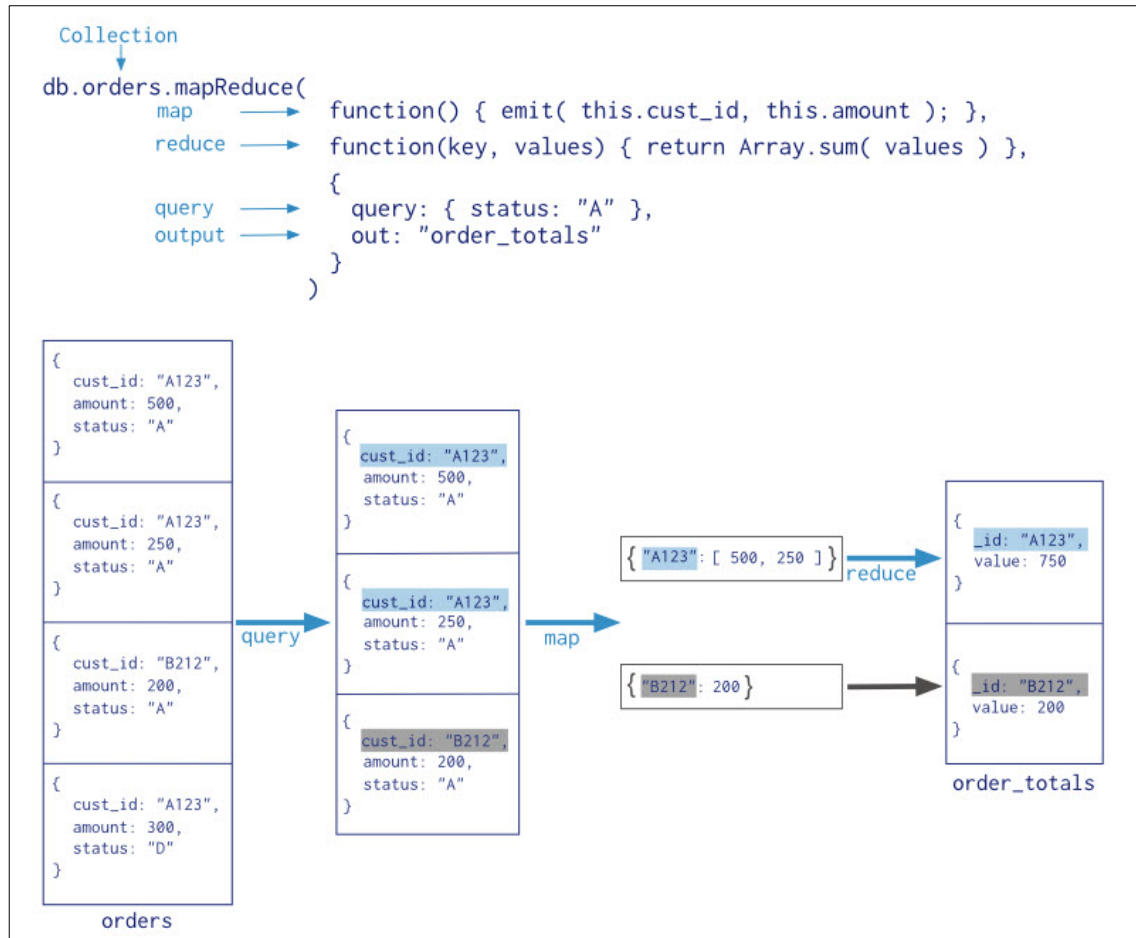
หลักการการทำงานของแมพรีดิวซ์จะถูกเรียกใช้งานวนซ้ำเรื่อยๆ จนกว่าจะได้ผลลัพธ์ที่เป็น Key-Value ชุดเดียว สำหรับผู้ที่ต้องการนำแมพรีดิวซ์ไปใช้ในการจัดการข้อมูลสามารถนำอัลกอริทึมของฟังก์ชันแมพและอัลกอริทึมของฟังก์ชันรีดิวซ์ไปใช้ในการเขียนโปรแกรมพัฒนาระบบได้ ในฟังก์ชันแมพสามารถใส่เงื่อนไขการตรวจสอบข้อมูลก่อนนำมาเข้าฟังก์ชันแมพได้

สำหรับแมพรีดิวซ์ในมอองโกดีบีนั้นใช้ภาษา Javascript โดยมีรูปแบบดังรูปที่ 2.8

Map function prototype	Reduce function prototype
<pre>function(){ ... emit(key,value); }</pre>	<pre>function(key,value){ ... return result; }</pre>

รูปที่ 2.8 รูปแบบของฟังก์ชันแมพและฟังก์ชันรีดิวซ์

จากรูปที่ 2.8 แสดงรูปแบบของฟังก์ชันแมพและฟังก์ชันรีดิวซ์ ภายในฟังก์ชันแมพผู้พัฒนาระบบสามารถที่จะเขียน โปรแกรมเพื่อกำหนดข้อมูลต่างๆก่อนที่จะส่งไปยังฟังก์ชันรีดิวซ์ในรูปแบบของ (key, value) ได้ เช่น ให้ส่งข้อมูลที่มีสถานะเป็น “A” เท่านั้น ส่วนฟังก์ชันรีดิวซ์จะเป็นการรับค่าข้อมูลจากฟังก์ชันแมพเพื่อมาประมวลผลตามความต้องการของผู้พัฒนาระบบ เช่น ให้รวมข้อมูลราคาที่มาจากผู้ซื้อคนเดียวกัน สำหรับการดำเนินงานของแมพรีดิวซ์แสดงในรูปที่ 2.9



รูปที่ 2.9 map-reduce operation [11]

จากรูปที่ 2.9 แสดงการทำงานของแมพรีดิวซ์ โดยมีข้อมูลอินพุตที่มาจากคอลเลกชัน `orders` ซึ่งมีการทำงานส่วนต่างๆดังนี้

- ส่วน `query` ทำหน้าที่กรองข้อมูลตามความต้องการ ตามตัวอย่างจะเป็นการกรองข้อมูลที่มี `status = "A"`
- ส่วน `map` ทำหน้าที่จัดส่งข้อมูลให้อยู่ในรูปของ `(key,value)` ซึ่งฟิลด์ที่อ้างถึงในฟังก์ชันแมพนี้จะมีคำว่า `"this"` นำหน้า หลังจากการทำงานของฟังก์ชันแมพเสร็จแต่ละ `key` จะมีค่า `value` หลายค่า
- ส่วน `reduce` ทำหน้าที่รวมและลดข้อมูล ตามตัวอย่างจะเป็นการรวมจำนวนราคาของค่า `value` ของแต่ละ `key`

ซึ่งจากรูปที่ 2.9 หากไม่ใช้กระบวนการของแมพรีดิวซ์ทำงานก็ต้องทำการเขียนโปรแกรมเพื่อประมวลผลซึ่งต้องอาศัยความสามารถของผู้พัฒนาระบบที่ต้องมีอัลกอริทึมในการเขียนโปรแกรมให้

มีประสิทธิภาพ อีกทั้งการเขียนโปรแกรมเพื่อประมวลผลเองต้องสูญเสียเวลาในการโอนถ่ายข้อมูลมาประมวลผลที่ระบบผู้ใช้งาน ซึ่งหากใช้แมพรีคิวซ์ทำงานก็จะลดเวลาในส่วนนี้ลงได้ทั้งยังให้การเขียนโปรแกรมทำได้ง่ายขึ้น

2.2 งานวิจัยที่เกี่ยวข้อง

ในการศึกษางานวิจัยที่เกี่ยวข้องนั้น ผู้วิจัยได้ทำการศึกษางานวิจัยที่พยายามเสนอแนวคิดในการเชื่อมความสัมพันธ์ของข้อมูลบนฐานข้อมูล โนเอสคิวแอล ซึ่งเป็นฐานข้อมูลที่ไม่มีคำสั่งสอบถามข้อมูลที่สามารถเชื่อมความสัมพันธ์ของข้อมูลที่แยกเก็บคนละส่วนได้ ผู้วิจัยได้คัดเลือกงานวิจัยที่เกี่ยวข้องและมีความใกล้เคียงกับงานวิจัยที่ดำเนินการอยู่ดังนี้

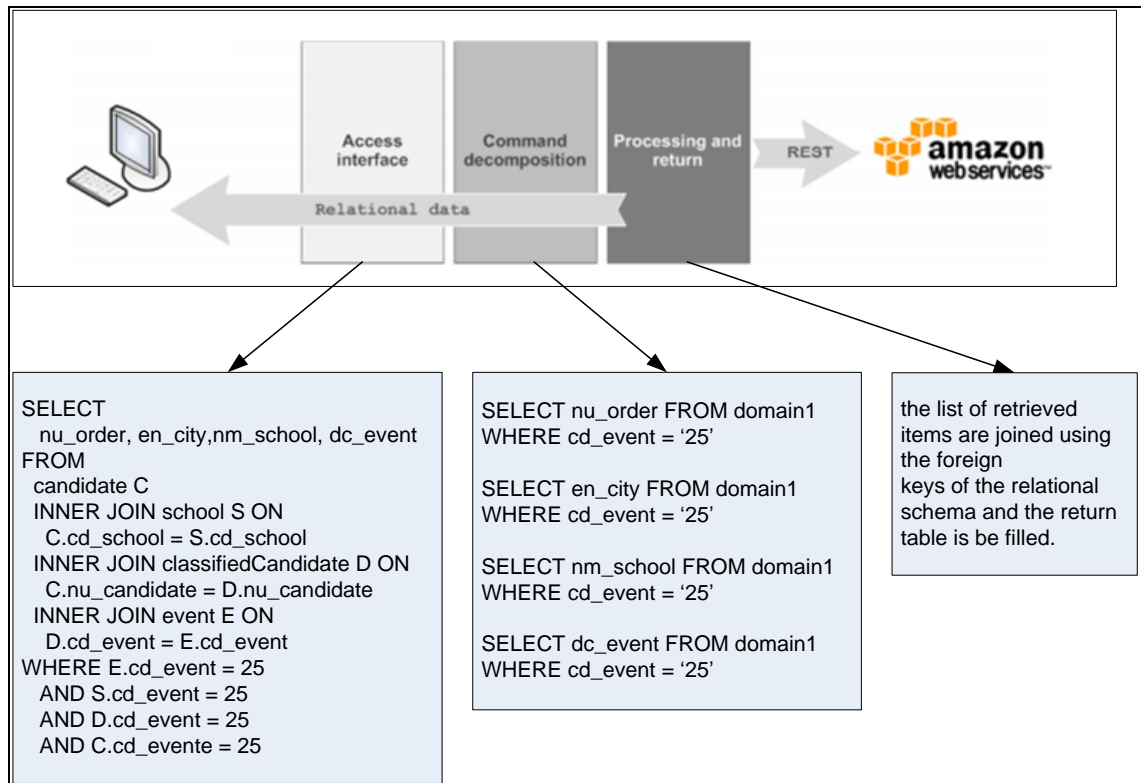
2.2.1 SimpleSQL: A Relational Layer for SimpleDB

งานวิจัย SimpleSQL: A Relational Layer for SimpleDB [12] นี้ได้นำเสนอวิธีการเชื่อมความสัมพันธ์ของข้อมูลบนฐานข้อมูลที่ชื่อว่า SimpleDB เพื่อใช้จัดการข้อมูลบนระบบคลาวด์ซึ่งเป็นฐานข้อมูล โนเอสคิวแอลแบบด็อกคิวเมนต์ชนิดหนึ่ง ด้วยวิธีการสร้างอินเทอร์เฟซที่ชื่อว่า SimpleSQL ขึ้นมาให้บริการเขียนคำสั่งด้วยภาษาเอสคิว โปรแกรมอินเทอร์เฟซนั้นมีการทำงาน 3 ส่วน ดังนี้

ส่วนที่ 1 Access interface ทำหน้าที่รับคำสั่งที่เป็นภาษาเอสคิวแอล เช่นคำสั่ง Insert, Update, Delete และคำสั่งเอสคิวแอลที่ใช้สำหรับเชื่อมความสัมพันธ์ระหว่างข้อมูล (Select) ซึ่งจะต้องเป็นแบบ INNER JOIN และไม่รองรับ sub queries การจัดเก็บข้อมูลในฐานข้อมูล SimpleDB นั้นจะต้องมีการออกแบบการเก็บข้อมูลให้มีการเชื่อมโยงระหว่างข้อมูลด้วยกันเพื่อรองรับการสอบถามข้อมูล

ส่วนที่ 2 Command decomposition ทำหน้าที่แยกคำสั่งเอสคิวแอล ซึ่งเป็นคำสั่งที่ใช้ดำเนินการกับข้อมูลที่รับเข้ามาให้เป็นคำสั่งที่อยู่ในรูปแบบของ SimpleDB โดยนำ regular expression เข้ามาใช้ในการตรวจสอบรูปแบบคำสั่งและใช้ในการแยกคำสั่งออกเป็นส่วนย่อยๆ

ส่วนที่ 3 Processing and Return ทำหน้าที่ส่งคำสั่งที่แยกออกเป็นส่วนย่อยๆส่งไปประมวลผลที่ฐานข้อมูล SimpleDB เมื่อได้ผลลัพธ์กลับมาแล้วจึงนำผลลัพธ์แต่ละส่วนมารวมกันเพื่อส่งกลับไปยังผู้ใช้ แต่ถ้าเป็นคำสั่ง Insert, Update และ Delete จะทำการส่งค่าที่เป็น Boolean กลับไปยังผู้เรียกใช้



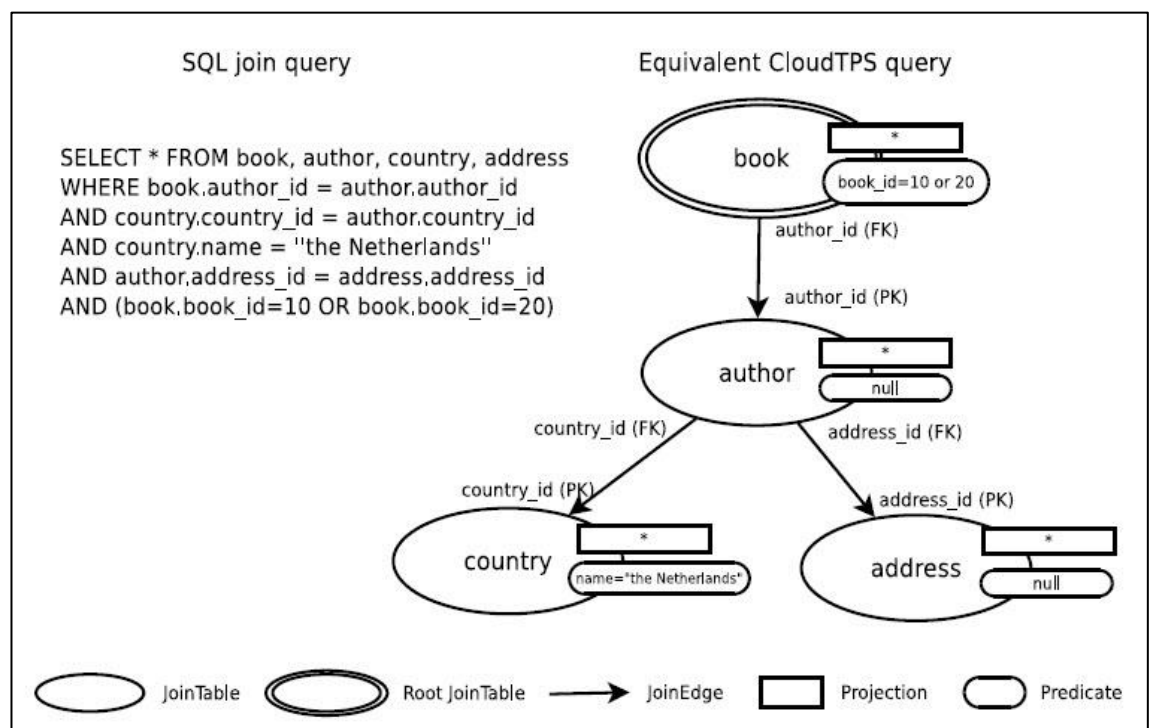
รูปที่ 2.10 Interface of SimpleSQL [12]

จากรูปที่ 2.10 แสดงตัวอย่างการทำงานของอินเตอร์เฟซที่ชื่อว่า SimpleSQL โดยส่วน Access Interface มีการรับคำสั่งที่เป็นภาษาเอสคิวแอลซึ่งมีลักษณะการเชื่อมโยงความสัมพันธ์ของตาราง ส่วน Command decomposition จะแยกคำสั่งเอสคิวแอลที่รับเข้ามาออกเป็นคำสั่งย่อยที่ตรงตามรูปแบบภาษาสอบถามที่ใช้ในฐานข้อมูล SimpleDB และส่วน Processing and return จะส่งคำสั่งย่อยทั้งสี่ไปประมวลผลที่ฐานข้อมูล SimpleDB แล้วนำผลลัพธ์ของแต่ละคำสั่งย่อยมาเชื่อมความสัมพันธ์โดยใช้คีย์นอก (Foreign key) ซึ่งพบว่าวิธีที่นำเสนอในงานวิจัยที่เกี่ยวข้องชิ้นนี้ค่อนข้างสูญเสียเวลาไปในส่วนของการโอนถ่ายข้อมูลเนื่องจากกระบวนการดังกล่าวต้องนำผลลัพธ์แต่ละส่วนจากฐานข้อมูลมาประมวลผลที่ฝั่งของระบบผู้ใช้งาน

2.2.2 Scalable Join Queries in Cloud Data Stores

งานวิจัย Scalable Join Queries in Cloud Data Stores [13] ที่เกี่ยวข้องชิ้นนี้ได้กล่าวถึงระบบคลาวด์ซึ่งเป็นระบบที่ไม่สนับสนุนในเรื่องของการเชื่อมความสัมพันธ์ของข้อมูล การสอบถามข้อมูลจะเป็นการสอบถามข้อมูลกับตารางเดียวโดยใช้คีย์หลัก (Primary Key) ในการเข้าถึงข้อมูล เพราะระบบคลาวด์นั้นหากมีการจัดเก็บข้อมูลแยกย่อยเป็นหลายตารางจะทำให้การขยายขนาดพื้นที่จัดเก็บข้อมูลและการจัดการข้อมูลทำได้ยากประกอบกับฐานข้อมูลส่วนใหญ่บนระบบคลาวด์นั้นเป็นฐานข้อมูลประเภท โน

เอสคิวแอลที่เน้นการจับเก็บข้อมูลที่มีความสัมพันธ์กันไว้ด้วยกันเพื่อให้ง่ายต่อการสอบถามข้อมูลงานวิจัยที่เกี่ยวข้องซึ่งนี้ได้แสดงให้เห็นถึงการออกแบบฐานข้อมูลให้แยกเก็บคนละส่วนเพื่อเชื่อมความสัมพันธ์กันในขณะเดียวกันก็ยังคงความสามารถในการขยายขนาดพื้นที่จัดเก็บข้อมูลบนระบบคลาวด์ได้ด้วย โดยผู้ทำงานวิจัยที่เกี่ยวข้องซึ่งนี้ได้กล่าวไว้ในระบบต่าง ๆ นั้น ผู้ใช้งานอาจจะไม่ได้ใช้ฟิลด์คีย์หลักในการค้นหาข้อมูลซึ่งฟิลด์นี้เรียกว่าคีย์รอง (Secondary Key) ซึ่งในระบบ CloudTPS ที่อ้างถึงในงานวิจัยนี้ จะสร้างตารางอินเด็กซ์ของฟิลด์ที่เป็นคีย์รองขึ้นมาคู่กับตารางหลัก เมื่อมีการปรับปรุงข้อมูลในตารางหลัก ตารางอินเด็กซ์ก็จะถูกปรับปรุงโดยอัตโนมัติ สำหรับการเชื่อมความสัมพันธ์ระหว่างตารางจะมีการพัฒนาโปรแกรมเป็นอินเตอร์เฟสขึ้นระหว่างระบบของผู้ใช้งานกับระบบ CloudTPS เพื่อใช้ในการเชื่อมความสัมพันธ์ของข้อมูล โดยอินเตอร์เฟสที่สร้างขึ้นต้องการข้อมูลที่ชื่อว่า JoinTable และ JoinEdge โดย JoinTable ประกอบด้วย ชื่อตาราง ชื่อฟิลด์ที่ต้องการและเงื่อนไข ส่วน JoinEdge ประกอบด้วย references ของทั้งสองตาราง



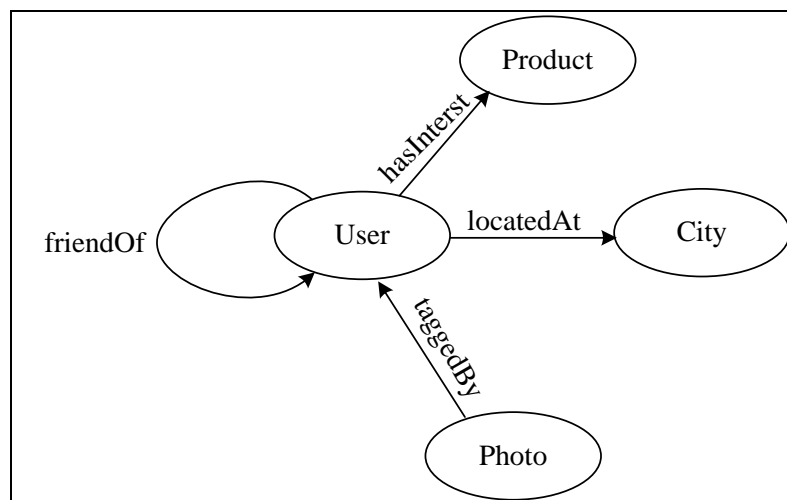
รูปที่ 2.11 CloudTPS's representation of a join query [13]

จากรูปที่ 2.11 แสดงถึงขั้นตอนในการปรับรูปแบบคำสั่งการเชื่อมความสัมพันธ์ของภาษาเอสคิวแอลไปยังรูปแบบการเชื่อมความสัมพันธ์ที่นำเสนอในงานวิจัยที่เกี่ยวข้องซึ่งนี้ ซึ่งกระบวนการเชื่อมความสัมพันธ์จะเริ่มต้นจาก Root table ซึ่ง Root table จะเป็นตารางในเงื่อนไขที่มีการระบุค่าของฟิลด์ที่เป็นคีย์หลัก (Primary key) และเป็นตารางที่ใช้คีย์นอก (Foreign key) อ้างอิงไปยังคีย์หลักของอีก

ตารางหนึ่ง ซึ่งในที่นี้ Root table คือ book table มีฟิลด์ book_id เป็นคีย์หลัก มีค่าคือ book_id=10 และ book_id=20 มีฟิลด์ author_id ซึ่งเป็นคีย์นอก ของตาราง “book” อ้างอิงไปยังฟิลด์ author_id ซึ่งเป็นคีย์หลักของตาราง “author” เงื่อนไขการเชื่อมความสัมพันธ์ของตาราง “country” จะเลือกข้อมูลที่มีผู้ประพันธ์มาจากประเทศ “Netherlands” แม้ว่างานวิจัยที่เกี่ยวข้องข้อนี้จะได้วัดประสิทธิภาพเรื่องของเวลาที่ใช้ในการเชื่อมความสัมพันธ์เพราะเน้นไปทางด้านความสามารถทางการขยายขนาด แต่ด้วยแนวคิดที่นำเสนอนี้แสดงให้เห็นว่าการเชื่อมความสัมพันธ์ระหว่างข้อมูลต้องดำเนินการที่ระบบผู้ใช้งานซึ่งทำให้เกิดการสูญเสียเวลาในการติดต่อข้อมูลระหว่างระบบผู้ใช้งานกับฐานข้อมูล

2.2.3 งานวิจัยที่ประยุกต์ใช้ MapReduce กับ SPARQL

ในงานวิจัย Efficient Social Network Data Query Processing on Mapreduce [14] ได้กล่าวถึงข้อมูลในโซเชียลเน็ตเวิร์กที่ส่วนมากจะแสดงในรูปแบบของ Resource Description Framework (RDF) โดยใช้ภาษา SPARQL ในการสอบถามข้อมูลและใช้แมพรีดิวซ์เป็นเครื่องมือในการวิเคราะห์ข้อมูล ซึ่งการแปลงคำสั่งสอบถามข้อมูลด้วยภาษา SPARQL ไปยังการทำงานของแมพรีดิวซ์นั้นไม่ค่อยมีประสิทธิภาพเท่าที่ควรเนื่องจากรูปแบบภาษาของ SPARQL ไม่เหมือนกับรูปแบบของภาษาเอสคิวแอล ดังนั้นจึงได้นำเสนอวิธีการแปลงคำสั่งสอบถามข้อมูลจากภาษา SPARQL ไปยังแมพรีดิวซ์ให้มีประสิทธิภาพมากขึ้น โดยมี 2 วิธีคือ Multiple-join-with-filter Primitive และ Select-Join Primitive ในขณะที่งานวิจัย SPARQL Basic Graph Pattern Processing with Iterative MapReduce [15] ได้นำเสนอการใช้ภาษา SPARQL ร่วมกับแมพรีดิวซ์ให้มีประสิทธิภาพมากขึ้น โดยการเลือกคีย์ที่ใช้เชื่อมความสัมพันธ์ที่ดี ซึ่งหลักการเลือกคีย์จะใช้วิธี Greedy selection และ Multiple selection ซึ่งทั้งสองงานวิจัยนี้ ต้องการที่จะนำแมพรีดิวซ์มาประยุกต์ใช้งานร่วมกับภาษา SPARQL



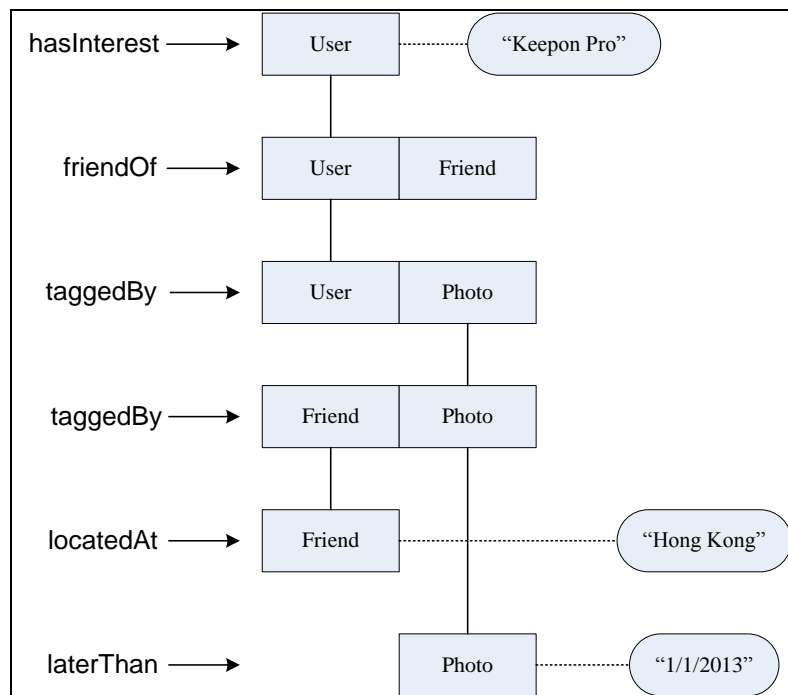
รูปที่ 2.12 Sample of RDF structure

จากรูปที่ 2.12 เป็นตัวอย่างรูปแบบการจัดเก็บข้อมูลอย่างง่ายของ RDF ซึ่งอยู่ในรูปแบบของกราฟ ที่ข้อมูลมีความสัมพันธ์กันแบบ Subject-Predicate-Object (set of triples) ซึ่งเป็นลักษณะของการเชื่อมโยงข้อมูลที่มีความสัมพันธ์กัน [16]

(?User hasInterest "Keepon Pro")	//Triple Pattern ID = 1
(?Friend locatedAt "Hong Kong")	//2
(?Photo laterThan "1/1/2013")	//3
(?User friendOf ?Friend)	//4
(?Photo taggedBy ?User)	//5
(?Photo taggedBy ?Friend)	//6

รูปที่ 2.13 Sample query of SPARQL [14]

จากรูปที่ 2.13 แสดงตัวอย่างเงื่อนไขที่ใช้ในการสอบถามข้อมูลของภาษา SPARQL ซึ่งประกอบไปด้วยหกเงื่อนไข โดยคำสั่งสอบถามข้อมูลของภาษา SPARQL จะอยู่ในรูปของ triples มีลักษณะคล้ายกับ RDF ยกเว้น subject, predicate และ object สามารถเป็นตัวแปรได้ [15] ซึ่งในกระบวนการทำงานของ SPARQL ร่วมกับแมพริคิวชันนั้นจะมีการสร้างตารางพื้นฐานขึ้นมาจำนวนหกตารางเท่ากับเงื่อนไข ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 ตารางพื้นฐานที่ถูกสร้างขึ้นจากเงื่อนไข

จากรูปที่ 2.14 แสดงตารางพื้นฐานที่ถูกสร้างขึ้นจากเงื่อนไขก่อนที่จะถูกเชื่อมความสัมพันธ์ ซึ่งคือที่ จะใช้เชื่อมความสัมพันธ์จะอยู่ในเส้นแนวตั้งที่เชื่อมต่อกันซึ่งจากรูปก็คือ User, Photo และ Friend

จากการศึกษา RDF ซึ่งเป็นโครงสร้างการเก็บข้อมูลที่มีความสัมพันธ์กัน ไปเรื่อยๆ โดยมีภาษา SPARQL ที่ใช้ในการสอบถามข้อมูล พบว่าการทำงานวิจัย Efficient Social Network Data Query Processing on Mapreduce [14] และ SPARQL Basic Graph Patten Processing with Iterative MapReduce [15] พยายามจะเชื่อมความสัมพันธ์ของข้อมูลด้วยการประยุกต์ใช้แมพรีดิวซ์นั้นต้องผ่าน กระบวนการสอบถามข้อมูลให้ได้เป็นชุดของข้อมูลก่อน เมื่อได้ชุดของข้อมูลแล้วจึงค่อยนำไป ประมวลผลในขั้นต่อไป

จากงานวิจัย SimpleSQL: A Relational Layer for SimpleDB [12] และ Scalable Join Queries in Cloud Data Stores [13] ที่ต้องสูญเสียเวลาในการประมวลผลระหว่างระบบผู้ใช้งานกับฐานข้อมูล ขณะที่งานวิจัย Efficient Social Network Data Query Processing on Mapreduce [14] และ SPARQL Basic Graph Patten Processing with Iterative MapReduce [15] ได้นำหลักการของแมพรีดิวซ์มา ประยุกต์ใช้เพื่อประมวลผลฝั่งฐานข้อมูลเพื่อลดการสูญเสียเวลาในการติดต่อระหว่างฐานข้อมูลกับ ระบบผู้ใช้งาน งานวิจัยชิ้นนี้จึงได้เสนอการเชื่อมความสัมพันธ์ของข้อมูลบนฐานข้อมูลมอดโกดึบี เนื่องจากภาษาที่ใช้ในการสอบถามข้อมูลของมอดโกดึบียังไม่สามารถเชื่อมความสัมพันธ์ข้อมูลที่อยู่ ต่างคอลเลกชันได้ ซึ่งไม่เหมือนกับงานวิจัย Efficient Social Network Data Query Processing on Mapreduce [14] และ SPARQL Basic Graph Patten Processing with Iterative MapReduce [15] เพราะรูปแบบการจัดเก็บข้อมูลของ RDF กับฐานข้อมูลมอดโกดึบีนั้นแตกต่างกัน ดังนั้นงานวิจัยนี้จึง ได้ใช้ฐานข้อมูลมอดโกดึบีเป็นกรณีศึกษาเนื่องจากฐานข้อมูลมอดโกดึบีนั้นเริ่มเป็นที่แพร่หลายของ ฐานข้อมูล โนเอสคิวแอลชนิดหนึ่ง

บทที่ 3

การใช้แมพรีดิคต์เชื่อมคอลเลคชันของฐานข้อมูลมองโกดีบี

ในงานวิจัยฉบับนี้ได้ทำการศึกษาและเปรียบเทียบการเชื่อมความสัมพันธ์ของคอลเลคชันของฐานข้อมูลมองโกดีบี โดยการเปรียบเทียบนั้นจะทำการเปรียบเทียบประสิทธิภาพรูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันได้แก่ รูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการเขียนโปรแกรมเชื่อมคอลเลคชันโดยตรง รูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการใช้แมพรีดิคต์และรูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการใช้แมพรีดิคต์ร่วมกับการสร้างคอลเลคชันชั่วคราว ซึ่งแต่ละรูปแบบจะมีวิธีการเชื่อมความสัมพันธ์ของคอลเลคชันได้สองวิธีคือ วิธีเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน A ไปคอลเลคชัน B และวิธีเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน B ไปคอลเลคชัน A ซึ่งการทำการเปรียบเทียบนั้น จะดำเนินการเปรียบเทียบในแง่ของเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลคชันของทั้งสามรูปแบบ

ในการออกแบบระบบเพื่อจัดเก็บข้อมูลโดยใช้ฐานข้อมูลโนเอสคิวแอลประเภทคอกิวเมนทั้นนั้นจะมีรูปแบบการจัดเก็บข้อมูลได้สองรูปแบบหลักคือรูปแบบ Embedded และ References ซึ่งขึ้นอยู่กับลักษณะของระบบและการใช้งาน

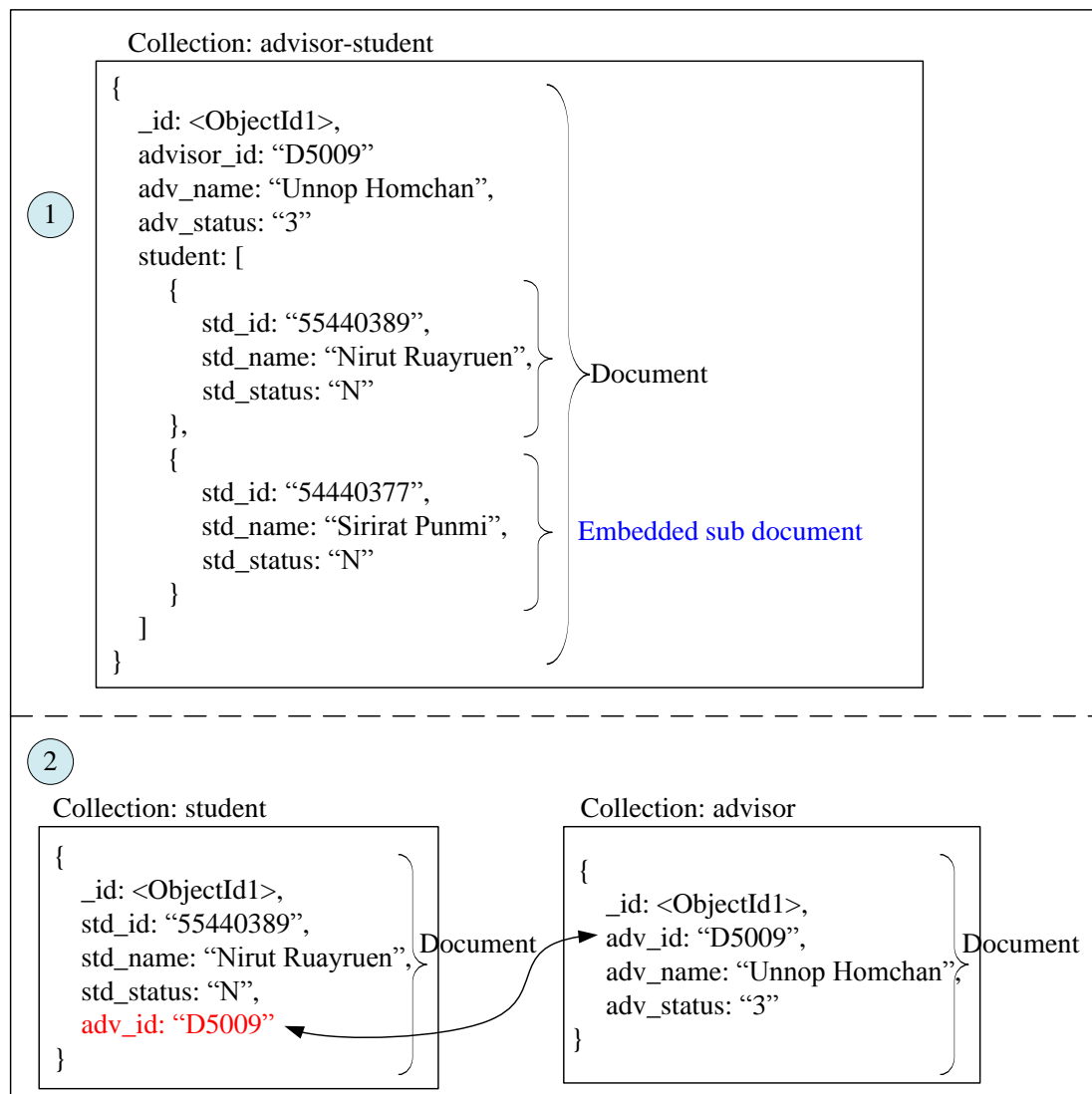
3.1 รูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลคชัน

การออกแบบเพื่อจัดเก็บข้อมูลของฐานข้อมูลมองโกดีบีนั้นสามารถแบ่งได้เป็นสองรูปแบบดังนี้คือ

- รูปแบบ Embedded Documents เป็นการจัดเก็บข้อมูลที่มีความสัมพันธ์กันให้อยู่ภายในคอกิวเมนต์และภายในคอลเลคชันเดียวกัน ตัวอย่างเช่น ความสัมพันธ์ระหว่างข้อมูลนักเรียนกับข้อมูลอาจารย์ที่ปรึกษา รูปแบบการจัดเก็บข้อมูลแบบ Embedded Documents จะทำการเก็บข้อมูลทั้งสองไว้ในคอกิวเมนต์ภายใต้คอลเลคชันเดียวกัน และจะทำการเก็บข้อมูลนักเรียนในรูปแบบคอกิวเมนต์ เราจะเรียกคอกิวเมนต์นี้เป็นคอกิวเมนต์หลักและภายในคอกิวเมนต์หลักนี้จะมีการเก็บข้อมูลอาจารย์ที่ปรึกษาในรูปแบบคอกิวเมนต์ที่ซ่อนภายในคอกิวเมนต์หลัก ซึ่งคอกิวเมนต์ที่เก็บข้อมูลอาจารย์ที่ปรึกษาจะเรียกว่า Sub Document
- รูปแบบ References คือการจัดเก็บข้อมูลที่มีความสัมพันธ์กันโดยการเก็บข้อมูลนั้นจะแบ่งเก็บข้อมูลทั้งสองไว้ต่างคอลเลคชันกัน แต่ละคอลเลคชันจะต้องมีจุดเชื่อมโยงหรือจุดอ้างอิงจากคอลเลคชันหนึ่งไปยังอีกคอลเลคชันหนึ่ง จุดที่เชื่อมความสัมพันธ์กันคือ

เรฟเฟอร์เรนซ์คีย์ (Reference key) ของคอลเลกชันหนึ่งเชื่อมโยงไปยังคีย์หลักของอีกคอลเลกชันหนึ่ง ตัวอย่างเช่น ความสัมพันธ์ระหว่างข้อมูลนักเรียนกับข้อมูลอาจารย์ที่ปรึกษา รูปแบบการจัดเก็บแบบ Reference จะทำการเก็บข้อมูลทั้งสองไว้ในต่างคอลเลกชันกัน โดยมีจุดเชื่อมโยงความสัมพันธ์กันคือ เรฟเฟอร์เรนซ์คีย์ของคอลเลกชันนักเรียนคือฟิลด์รหัสอาจารย์ที่ปรึกษาเชื่อมโยงกับคีย์หลักของคอลเลกชันอาจารย์ที่ปรึกษาคือฟิลด์รหัสอาจารย์ที่ปรึกษา

รูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันที่กล่าวมานั้น สามารถดูตัวอย่างการจัดเก็บข้อมูลของสองรูปแบบได้ดังรูปที่ 3.1



รูปที่ 3.1 รูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลกชัน

จากรูปที่ 3.1 แสดงรูปแบบการจัดเก็บข้อมูลเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันในฐานข้อมูลมองโกดีบี โดยมีสองรูปแบบ คือรูปแบบที่หนึ่ง เป็นการออกแบบข้อมูลที่มีความสัมพันธ์กันให้อยู่ซ้อนกันเป็นลำดับชั้น โดยออกแบบให้มีคอคิวเมนต์หลักแล้วมีคอคิวเมนต์ย่อยซ้อนอยู่ในภายใต้คอลเลกชันเดียวกัน ซึ่งวิธีการนี้จะมีประสิทธิภาพมากในเรื่องของการอ่านข้อมูลเพราะสามารถอ่านข้อมูลที่มีความสัมพันธ์กันได้ทันทีโดยไม่ต้องเสียเวลาในการเชื่อมความสัมพันธ์ของข้อมูลแต่ก็มีข้อเสียคือถ้าต้องการข้อมูลเฉพาะของนักเรียนก็ต้องเสียเวลาเข้าถึงข้อมูลอาจารย์ที่ปรึกษา ก่อน และหากข้อมูลนักเรียนมีการเปลี่ยนแปลงอาจารย์ที่ปรึกษาใหม่ก็ต้องทำการลบข้อมูลนักเรียนออกจากอาจารย์ที่ปรึกษาคนเดิมก่อนแล้วจึงทำการเพิ่มข้อมูลนักเรียนไปยังอาจารย์ที่ปรึกษาคนใหม่ซึ่งต้องเสียเวลาการทำงานถึงสองขั้นตอน ส่วนรูปแบบที่สองเป็นการออกแบบข้อมูลที่มีความสัมพันธ์กันแยกเก็บคนละคอลเลกชัน ซึ่งภายในคอคิวเมนต์ของคอลเลกชันนักเรียนจะมีฟิลด์ที่ใช้อ้างอิงไปยังคอคิวเมนต์ของคอลเลกชันอาจารย์ที่ปรึกษา การออกแบบในรูปแบบนี้เพื่อให้การปรับปรุงข้อมูลทำได้รวดเร็วขึ้น เช่น หากนักเรียนมีการเปลี่ยนแปลงอาจารย์ที่ปรึกษาก็เพียงแค่เปลี่ยนรหัสอาจารย์ที่ปรึกษาในคอลเลกชันของนักเรียนซึ่งไม่เหมือนกับรูปแบบที่หนึ่งที่จะต้องลบข้อมูลนักเรียนออกจากอาจารย์ที่ปรึกษาคนเดิมก่อนแล้วจึงไปเพิ่มข้อมูลนักเรียนกับอาจารย์ที่ปรึกษาคนใหม่ แต่จากการจัดเก็บข้อมูลในรูปแบบที่สองนี้ทำให้ข้อมูลที่มีความสัมพันธ์กันถูกจัดเก็บไว้คนละคอลเลกชัน ทำให้การนำข้อมูลไปใช้งานมีความจำเป็นต้องเขียนคำสั่งเพื่อสอบถามข้อมูลที่ละคอลเลกชันเช่น หากต้องการทราบว่านิสิตมีอาจารย์ที่ปรึกษาเป็นใคร จะต้องทำการเขียนโปรแกรมเพื่อสอบถามข้อมูลไปยังคอลเลกชันนักเรียนก่อนเมื่อได้ข้อมูลมาแล้วจึงนำรหัสอาจารย์ที่ปรึกษาของคอลเลกชันนักเรียนไปสอบถามข้อมูลจากคอลเลกชันอาจารย์ที่ปรึกษาเพื่อให้ทราบข้อมูลอาจารย์ที่ปรึกษา เนื่องจากฐานข้อมูลมองโกดีบีไม่มีชุดคำสั่งสอบถามข้อมูลที่ประมวลผลการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่คนละคอลเลกชันได้ ดังนั้นงานวิจัยนี้จึงได้ทำการศึกษาหาแนวทางการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่คนละคอลเลกชันให้เกิดขึ้นและทดสอบประสิทธิภาพในการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่คนละคอลเลกชัน จึงได้ทำการศึกษารูปแบบการเชื่อมความสัมพันธ์ของข้อมูลจำนวน 3 รูปแบบ ดังนี้

- รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง
- รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิคซ์
- รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิคซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

3.2 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง

การเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมเพื่อเชื่อมความสัมพันธ์ข้อมูลที่เก็บในฐานข้อมูลมองโกดีบีโดยตรงนั้น จะเป็นในลักษณะที่ผู้พัฒนาระบบเป็นคนเขียนโปรแกรมเพื่อจัดการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่คนละคอลเลกชันด้วยตนเอง โดยผู้พัฒนาระบบต้องดำเนินการ

เขียนโปรแกรมเพื่อส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชัน A เพื่อให้ได้ข้อมูลและค่าเรฟเฟอร์เรนซ์คีย์ที่เก็บในคอลเลกชัน A หลังจากนั้นต้องส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชัน B อีกครั้งโดยใช้เรฟเฟอร์เรนซ์คีย์ที่ได้จากคอลเลกชัน A เป็นเงื่อนไขในการค้นหาข้อมูลที่ต้องการ อัลกอริทึมที่ใช้การเชื่อมความสัมพันธ์ของรูปแบบนี้มีหลักการดังรูปที่ 3.2

```

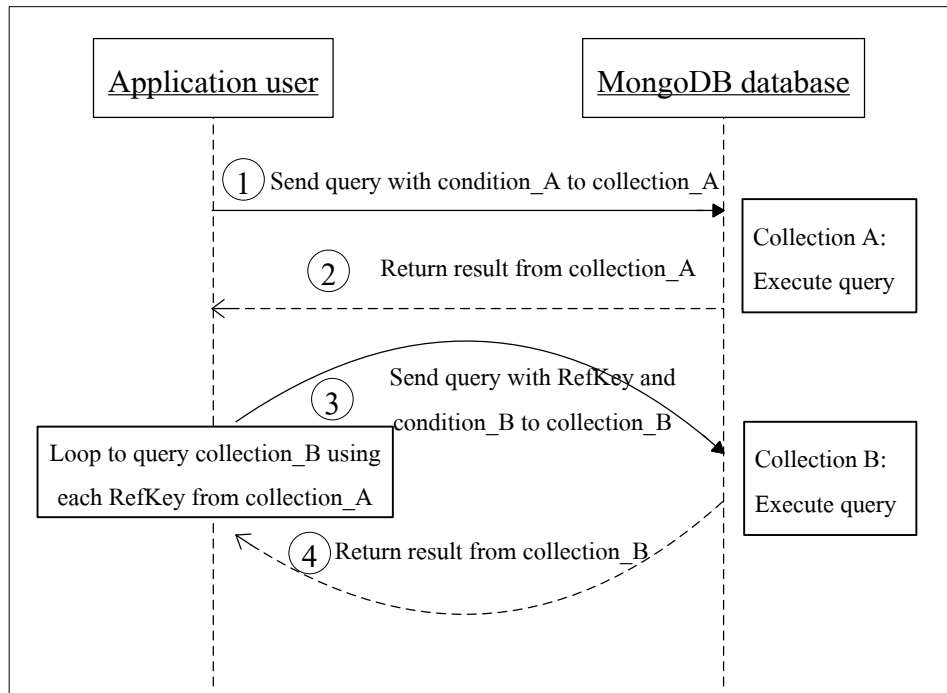
Algorithm : Direct join
BEGIN
1: dataset x = get data with condition from collection_A
2: for all data in dataset x do
3:   read collection_A.field-1,
      ...,
      read collection_A.field-k,
4:   read reference key /* for lookup in collection_B */
5:   rs = lookup in collection_B use reference key and condition of collection_B
6:   for all i in rs do
7:     read collection_B.field-1,
      ...,
      read collection_B.field-k
8:   end for
9: end for
END

```

รูปที่ 3.2 แสดงอัลกอริทึมเชื่อมคอลเลกชันจากการเขียนโปรแกรมโดยตรง

จากรูปที่ 3.2 แสดงอัลกอริทึมในการเชื่อมคอลเลกชันจากการเขียนโปรแกรมโดยตรงด้วยวิธีเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B จากรูปในบรรทัดที่ 1 เป็นการส่งคำสั่งพร้อมระบุเงื่อนไขไปสอบถามข้อมูลที่อยู่คอลเลกชัน A จากนั้นในบรรทัดที่ 2-4 เป็นการวนลูปเพื่ออ่านค่าฟิลด์ของข้อมูลที่ต้องการจากคอลเลกชัน A จากนั้นในบรรทัดที่ 5 เป็นการส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชัน B โดยใช้ค่าเรฟเฟอร์เรนซ์คีย์จากบรรทัดที่ 4 รวมกับเงื่อนไขของคอลเลกชัน B ในส่วนสุดท้าย บรรทัดที่ 6-8 เป็นการวนลูปเพื่ออ่านค่าฟิลด์ของข้อมูลที่ต้องการจากคอลเลกชัน B

สำหรับขั้นตอนของรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรงสามารถอธิบายกระบวนการได้ดังรูปที่ 3.3 ดังนี้



รูปที่ 3.3 แสดงการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง

ในรูปที่ 3.3 แสดงขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรงด้วยวิธีเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B โดยมีขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1 ระบบของผู้ใช้งานส่งคำสั่งสอบถามข้อมูลพร้อมระบุเงื่อนไขที่ต้องการไปยังฐานข้อมูลมองโกดีบี โดยจะเริ่มสอบถามข้อมูลที่อยู่ในคอลเลกชัน A ก่อน

ขั้นตอนที่ 2 ฐานข้อมูลมองโกดีบีจะดำเนินการประมวลผลชุดคำสั่งสอบถามข้อมูลที่ได้รับมาและส่งผลลัพธ์ที่ได้กลับไปยังระบบของผู้ใช้งาน ซึ่งผลลัพธ์ที่ได้มาจะเป็นชุดของข้อมูล (dataset) ที่มีเฉพาะข้อมูลที่ตรงตามเงื่อนไขของคอลเลกชัน A

ขั้นตอนที่ 3 เมื่อโปรแกรมภายในระบบผู้ใช้งานได้รับค่าผลลัพธ์จากฐานข้อมูลมองโกดีบีที่มาจากคอลเลกชัน A แล้ว ภายในโปรแกรมจะทำการวนลูปอ่านค่าข้อมูลของฟิลด์ที่ต้องการและมีการอ่านค่าเรฟเฟอร์เรนซ์คีย์โดยนำค่าเรฟเฟอร์เรนซ์คีย์ร่วมกับเงื่อนไขของคอลเลกชัน B เพื่อเป็นเงื่อนไขของคำสั่งที่ใช้ในการสอบถามข้อมูลของคอลเลกชัน B ส่งไปยังฐานข้อมูลมองโกดีบี

ขั้นตอนที่ 4 ฐานข้อมูลมองโกดีบีจะดำเนินการประมวลผลคำสั่งสอบถามข้อมูลและส่งผลลัพธ์กลับไปยังระบบของผู้ใช้งาน

3.3 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์

งานวิจัยนี้ได้นำเสนอวิธีการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่คนละคอลเลกชันของฐานข้อมูลมอโกดีบี โดยนำเสนอรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันที่นำวิธีการเขียนโปรแกรมแบบแมพรีดิวซ์มาประยุกต์ใช้เพื่อเชื่อมความสัมพันธ์ของข้อมูลที่อยู่ต่างคอลเลกชัน เพื่อลดขั้นตอนในการเขียนโปรแกรมส่งคำสั่งไปสอบถามข้อมูลที่ละคอลเลกชัน สำหรับอัลกอริทึมแมพรีดิวซ์ที่นำมาประยุกต์ใช้เชื่อมคอลเลกชันแสดงได้ดังรูปที่ 3.4

<p>Algorithm : Map function Input: dataset x e collection_A BEGIN 1: for all data in dataset x do 2: rs = lookup in collection_B use RefKey from x and condition of collection_B 3: for all i in rs do 4: value = { collection_A.field-1, ..., collection_A.field-k, collection_B.field-1(i), ..., collection_B.field-k(i) } 5: emit(ObjectID,value) 6: end for 7: end for END</p>
<p>Algorithm : Reduce function Input: (([id,value-1), (id,value-2), ..., (id,value-k)]) BEGIN 1: output = { field-1, field-2, ..., field-k } 2: for all j in input list do 3: output.field-1 = value(j).field-1, ..., output.field-k = value(j).field-k 4: end for 5: return output END</p>
<p>Algorithm : Invoke MapReduce BEGIN 1: call collection_A.mapReduce(mapFunction, reduceFunction, { query: { condition of collection_A } }) END</p>

รูปที่ 3.4 อัลกอริทึมเชื่อมคอลเลกชันจากการประยุกต์ใช้เทคนิคแมพรีดิวซ์

จากรูปที่ 3.4 แสดงอัลกอริทึมในการเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B ด้วยการประยุกต์ใช้แมพรีดิวซ์ ประกอบไปด้วย 3 อัลกอริทึมดังนี้

- อัลกอริทึม “Map function” มีอินพุตคือชุดของข้อมูลที่เกิดจากเงื่อนไขของอัลกอริทึม “Invoke MapReduce” โดยกระบวนการทำงานจะเริ่มจากบรรทัดที่ 2 เป็นการค้นหาข้อมูลที่อยู่ในคอลเลกชัน B โดยการใช้ค่าเรเฟอเรนซ์คีย์จากข้อมูลอินพุตรวมกับเงื่อนไขของคอลเลกชัน B หลังจากนั้นเป็นการสร้างตัวแปรขึ้นมารองรับค่าฟิลด์ที่ต้องการซึ่งมีทั้งฟิลด์ที่มาจากคอลเลกชัน A และคอลเลกชัน B ในส่วนของบรรทัดที่ 5 เป็นการส่งข้อมูลให้อัลกอริทึม “Reduce function” ซึ่งเป็นฟังก์ชันรีดิวซ์ตามรูปแบบ (คีย์, แวลู) ซึ่งค่าคีย์ในที่นี้คือค่า ObjectID เป็นค่าที่ฐานข้อมูลมองโกดีบีสร้างให้ และค่าแวลูคือค่าของชุดข้อมูลที่มาจกคอลเลกชัน A และคอลเลกชัน B
- อัลกอริทึม “Reduce function” มีอินพุตคือชุดของ (คีย์, แวลู) ที่มาจากอัลกอริทึม “Map function” ซึ่งเป็นฟังก์ชันแมพ โดยบรรทัดที่ 1 จะเป็นการสร้างตัวแปรที่เป็นเซตของฟิลด์ที่ต้องการ หลังจากนั้นเป็นการกำหนดค่าให้กับฟิลด์ที่สร้างขึ้นจากค่าแวลู และบรรทัดที่ 5 เป็นการรีเทิร์นเอาต์พุตจากการเชื่อมคอลเลกชัน A กับคอลเลกชัน B
- อัลกอริทึม “Invoke MapReduce” เป็นอัลกอริทึมในการเรียกใช้งานแมพรีดิวซ์ให้ทำงานโดยมีค่าพารามิเตอร์คือฟังก์ชันแมพ ฟังก์ชันรีดิวซ์และเงื่อนไขของคอลเลกชัน A

วิธีที่นำเสนอในงานวิจัยนี้ได้มีการพัฒนาโปรแกรมให้เป็นไลบรารีเพื่อใช้เชื่อมความสัมพันธ์ของคอลเลกชันซึ่งในการพัฒนาไลบรารีที่ใช้ในการเชื่อมความสัมพันธ์ของข้อมูลที่ถูกจัดเก็บในคอลเลกชัน A และคอลเลกชัน B นั้นทำให้ผู้ที่ต้องการข้อมูลที่มีความสัมพันธ์กันของทั้งสองคอลเลกชันสามารถดึงข้อมูลโดยการเรียกใช้งานไลบรารีที่สร้างขึ้นก็จะสามารถได้รับข้อมูลที่ต้องการโดยไม่ต้องเขียนโปรแกรมเพื่อดึงข้อมูลที่อยู่ในทั้งสองคอลเลกชันเอง ซึ่งเป็นผลดีต่อนักพัฒนาระบบที่ไม่จำเป็นต้องเขียนอัลกอริทึมดึงข้อมูลมาเชื่อมความสัมพันธ์เอง การเขียนโปรแกรมเพื่อดึงข้อมูลที่มีความสัมพันธ์กันแต่เก็บค่าไว้ในต่างคอลเลกชันกันนั้นอาจทำให้สูญเสียเวลาในการประมวลผลเพราะต้องมีการใช้อัลกอริทึมในการวนลูปเพื่อดึงข้อมูลจากฐานข้อมูลมองโกดีบีและยังทำให้ประสิทธิภาพของเซิร์ฟเวอร์ฐานข้อมูลมองโกดีบีมีประสิทธิภาพการทำงานลดลงเพราะถ้ามีข้อมูลที่เก็บไว้ในแต่ละคอลเลกชันที่มีจำนวนมาก การวนลูปเพื่อดึงข้อมูลต้องใช้เวลานานประกอบกับถ้ามีผู้ใช้งานระบบจำนวนมากพร้อมกันยิ่งส่งผลต่อประสิทธิภาพการทำงานของเซิร์ฟเวอร์ฐานข้อมูลมองโกดีบีมากยิ่งขึ้น

3.3.1 ข้อกำหนดของไลบรารีการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์

ในการพัฒนาไลบรารีจากรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ได้มีการออกแบบให้ไลบรารีรับค่าพารามิเตอร์ 4 ค่า ซึ่งมีชื่อและรูปแบบการเรียกใช้งาน ดังรูปที่ 3.5

Parameter 1 : sSelect	□ collection_name_1, collection_name_2, ..., collection_name_n
Parameter 2 : sFrom	□ collection_name_1, collection_name_2
Parameter 3 : sWhere	□ collection_name_1.key_id, collection_name_2.key_id
Parameter 4 : sCondition	□ collection_name.field < comparison operator > value

รูปที่ 3.5 ข้อกำหนดของไลบรารีเพื่อเชื่อมความสัมพันธ์ของคอลเลกชัน

จากรูปที่ 3.5 เป็นข้อกำหนดสำหรับส่งค่าพารามิเตอร์ให้กับไลบรารีที่พัฒนาขึ้น โดยมีความหมาย ดังนี้

พารามิเตอร์ค่าที่ 1 sSelect เป็นค่าของชื่อฟิลด์ข้อมูลที่ต้องการสอบถาม มีรูปแบบคือ collection_name.field_1, collection_name.field_2,, collection_name.field_n ตัวอย่างของค่าข้อมูลที่ได้รับค่ามาดังนี้ เช่น student.std_id, student.std_title_th, student.std_name, student.surname, advisor.first_name, advisor.last_name เป็นต้น

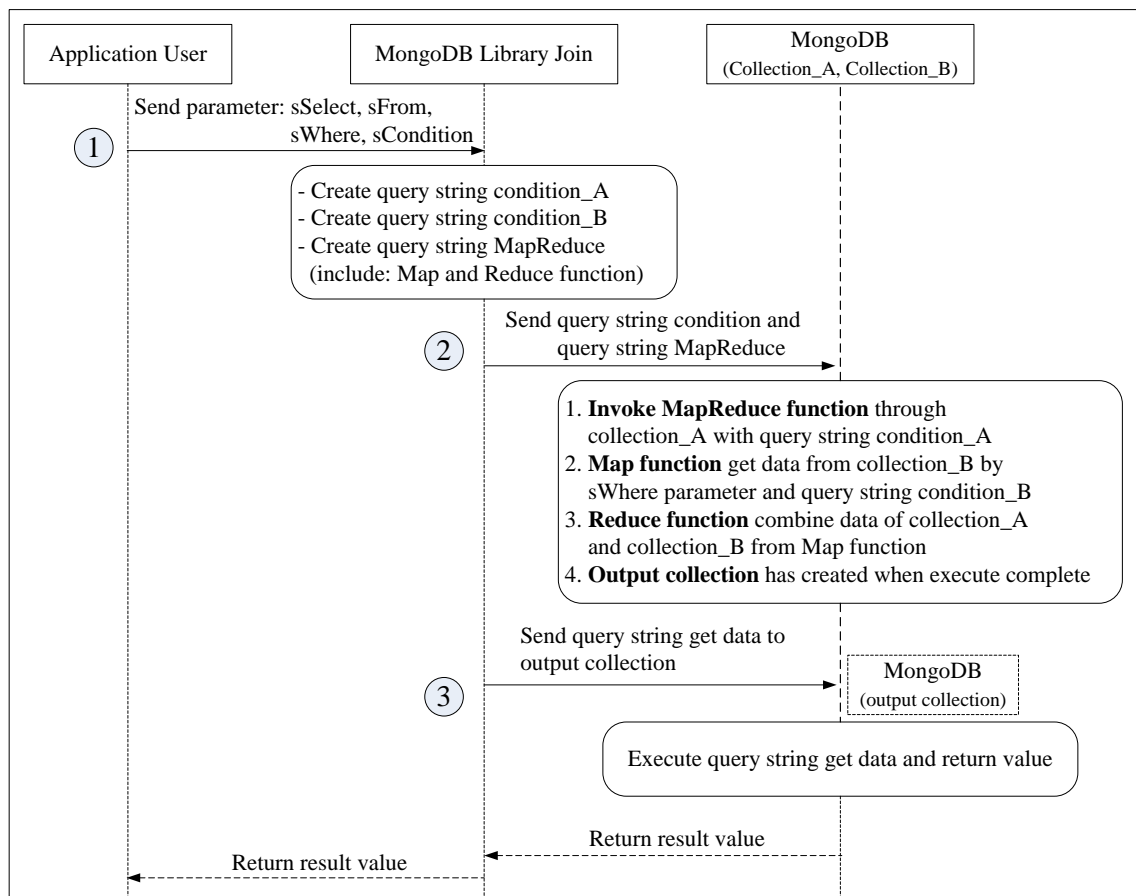
พารามิเตอร์ค่าที่ 2 sFrom เป็นค่าของชื่อคอลเลกชันที่ต้องการเชื่อมความสัมพันธ์ของข้อมูลเข้าด้วยกัน มีรูปแบบคือ collection_name_1, collection_name_2 ตัวอย่างของค่าข้อมูลที่ได้รับมา เช่น student, advisor เป็นต้น

พารามิเตอร์ค่าที่ 3 sWhere เป็นค่าที่ระบุชื่อฟิลด์ที่เป็นคีย์เพื่อใช้เชื่อมความสัมพันธ์ของคอลเลกชัน มีรูปแบบคือ collection_name_1.key_id = collection_name_2.key_id ตัวอย่างของค่าข้อมูลที่ได้รับมา เช่น student.advisor_id = advisor.advisor_id เป็นต้น

พารามิเตอร์ค่าที่ ค่าที่ 4 sCondition เป็นค่าของเงื่อนไขในการสอบถามข้อมูล มีรูปแบบคือ Collection_name.field <Comparison Operator> value, collection_name.field <Comparison Operator> value ตัวอย่างของค่าข้อมูลที่ได้รับมา เช่น student.extra_status='N', student.admit_proj=52, student.major_id='Q03', advisor.adv_status=3 เป็นต้น ซึ่ง <Comparison Operator> ได้แก่ =, <, >, <=, >= และ !=

3.3.2 ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์

การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์นั้น ได้นำเสนอวิธีการเขียนโปรแกรมสร้างชุดคำสั่งแมพรีดิวซ์ให้เป็นไลบรารี โดยไลบรารีจะทำการสร้างชุดคำสั่งแมพรีดิวซ์ตามค่าพารามิเตอร์ที่ส่งมาจากผู้ใช้งานระบบ แล้วส่งชุดคำสั่งนี้ไปประมวลผลที่ฐานข้อมูลมองโกดีบี สำหรับไลบรารีที่พัฒนาขึ้นเพื่อให้ง่ายในการอ้างอิงได้ใช้ชื่อว่า MongoDB Library Join โดยขั้นตอนการทำงานเชื่อมความสัมพันธ์ของคอลเลกชันอธิบายได้ดังต่อไปนี้



รูปที่ 3.6 ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้เทคนิคแมพรีดิวซ์

จากรูปที่ 3.6 สามารถอธิบายขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้เทคนิคแมพรีดิวซ์ได้ดังนี้

ขั้นตอนที่ 1 ระบบของผู้ใช้งานเรียกใช้ไลบรารี MongoDB Library Join โดยส่งค่าพารามิเตอร์ตามที่ไลบรารีต้องการ คือ sSelect, sFrom, sWhere และ sCondition ไลบรารี MongoDB Library Join เมื่อได้รับค่าข้อมูลตามพารามิเตอร์ที่ระบุ จะนำค่าข้อมูลจากพารามิเตอร์ sCondition มาสร้าง query string condition ซึ่งเป็นเงื่อนไขในการสอบถามข้อมูล โดยสร้างให้อยู่ใน

รูปแบบคำสั่งสอบถามที่ใช้ในฐานข้อมูลมอุงโกดีบี กระบวนการต่อมาจะนำค่าข้อมูลจากพารามิเตอร์ sSelect, sFrom และ sWhere พร้อมกับ query string condition มาสร้างเป็น query string MapReduce

ขั้นตอนที่ 2 ต่ query string condition และ query string MapReduce ไปประมวลผลที่ฐานข้อมูลมอุงโกดีบี โดย query string MapReduce จะเรียกใช้งานผ่านคอลเลกชัน A ที่มี query string condition_A เพื่อกรองเฉพาะข้อมูลที่ต้องการ ต่อมาฟังก์ชันแมพจะส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชัน B โดยมีค่าพารามิเตอร์ sWhere จากคอลเลกชัน A รวมกับค่าเงื่อนไขจาก query string condition_B โดยภายในฟังก์ชันแมพจะส่งข้อมูลไปยังฟังก์ชันรีดิวซ์ในรูปแบบ (คีย์, แวลู) ซึ่งในที่นี้ค่าคีย์เป็นค่าที่ไลบรารีสร้างให้ส่วนค่าแวลูเป็นค่าที่เกิดจากข้อมูลของคอลเลกชัน A และคอลเลกชัน B ต่อมาฟังก์ชันรีดิวซ์จะทำการรวมข้อมูลที่มาจากคอลเลกชัน A และคอลเลกชัน B ให้ตรงกับชื่อฟิลด์ที่ต้องการ เมื่อการทำงานของแมพรีดิวซ์ทำงานเสร็จจะมีการสร้างคอลเลกชันเอาต์พุตขึ้นมา

ขั้นตอนที่ 3 ไลบรารี MongoDB Library Join จะส่งคำสั่งสอบถามข้อมูลไปที่คอลเลกชันเอาต์พุตตามค่าพารามิเตอร์ sSelect เพื่อส่งผลลัพธ์กลับไปยังระบบผู้ใช้งาน

จากรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์นั้นพบว่าการใช้แมพรีดิวซ์นั้นจะเรียกใช้ผ่านคอลเลกชันใดคอลเลกชันหนึ่งและสามารถที่จะส่งเงื่อนไขของคอลเลกชันนั้นเพื่อกรองข้อมูลเฉพาะที่ต้องการได้ แต่คอลเลกชันที่ไม่ถูกเรียกใช้โดยแมพรีดิวซ์จะไม่สามารถกรองข้อมูลก่อนการเชื่อมคอลเลกชันได้ ดังนั้นงานวิจัยนี้จึงได้เสนอให้มีการสร้างคอลเลกชันชั่วคราวของคอลเลกชันที่ไม่ถูกเรียกใช้งานโดยแมพรีดิวซ์ เพื่อเป็นการกรองข้อมูลเฉพาะที่ต้องการมาใช้ในการกระบวนการเชื่อมคอลเลกชันเท่านั้น ดังแสดงในหัวข้อที่ 3.4

3.4 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

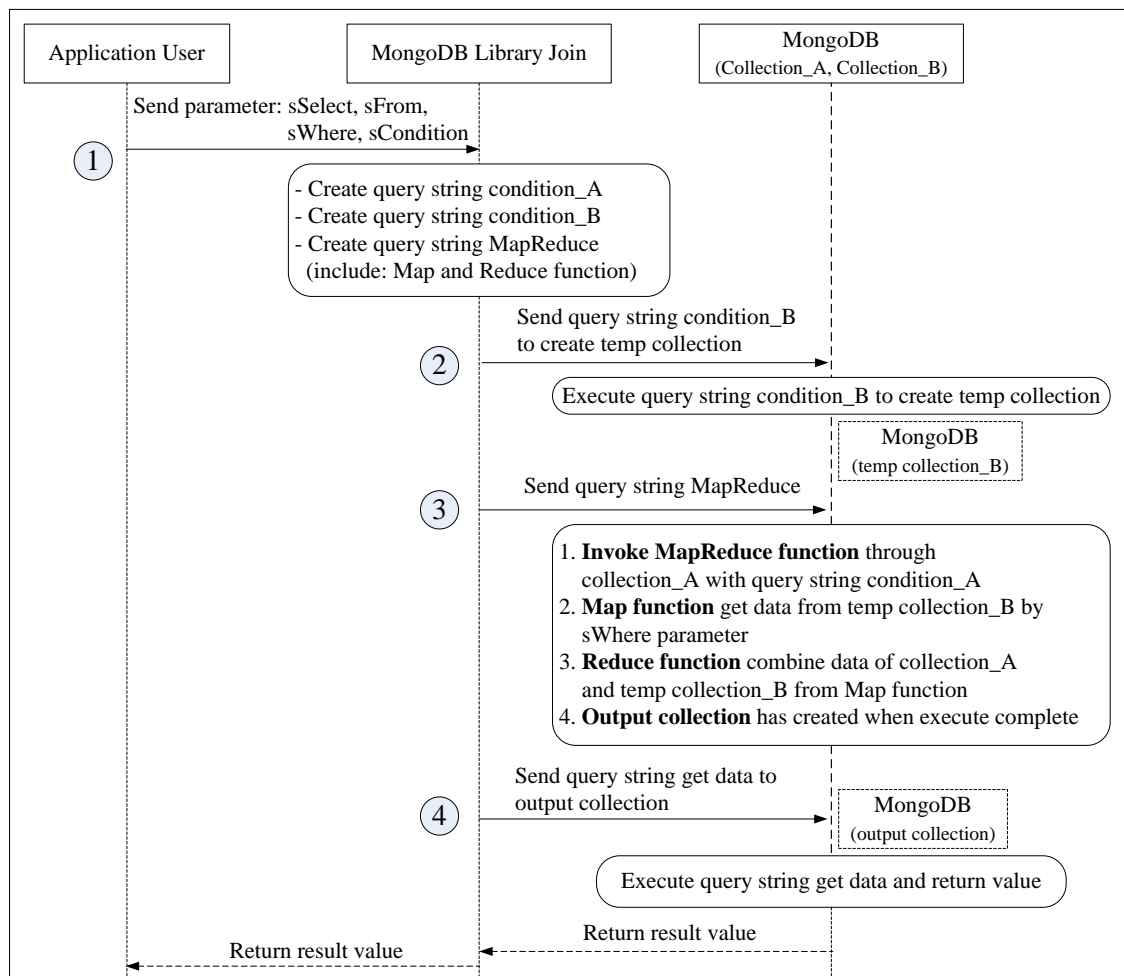
การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราวนั้นจะคล้ายคลึงกับหัวข้อ 3.3 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์เพียงแต่ไลบรารีที่สร้างตามหัวข้อนี้จะมีกระบวนการสร้างคอลเลกชันชั่วคราวขึ้นมาก่อน โดยการสร้างคอลเลกชันชั่วคราวจะสร้างจากเงื่อนไขของข้อมูลที่อยู่ในคอลเลกชันที่ไม่ถูกเรียกโดยแมพรีดิวซ์ (คอลเลกชัน B) เมื่อเข้าสู่กระบวนการเชื่อมคอลเลกชัน ชุดคำสั่งแมพรีดิวซ์จะทำงานระหว่างคอลเลกชัน A กับคอลเลกชันชั่วคราวที่สร้างขึ้น (คอลเลกชัน B)

3.4.1 ข้อกำหนดของไลบรารีการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

ในการพัฒนาไลบรารีจากรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว ได้มีการออกแบบให้ไลบรารีรับค่าพารามิเตอร์ 4 ค่า ซึ่งมีชื่อและรูปแบบการเรียกใช้งานเหมือนกับหัวข้อ 3.3.1

3.4.2 ขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราวนั้น ได้นำเสนอวิธีการเขียนโปรแกรมสร้างชุดคำสั่งแมพรีดิวซ์ให้เป็นไลบรารีโดยไลบรารีที่สร้างขึ้นี้จะมีกระบวนการสร้างคอลเลกชันชั่วคราวเพิ่มขึ้น สำหรับไลบรารีที่พัฒนาขึ้นได้ใช้ชื่อว่า MongoDB Library Join โดยขั้นตอนการทำงานเชื่อมความสัมพันธ์ของคอลเลกชันอธิบายได้ดังต่อไปนี้



รูปที่ 3.7 การเชื่อมความสัมพันธ์คอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

จากรูปที่ 3.7 สามารถอธิบายขั้นตอนการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ ร่วมกับการสร้างคอลเลกชันชั่วคราวได้ดังนี้

ขั้นตอนที่ 1 ระบบของผู้ใช้งานเรียกใช้ไลบรารี MongoDB Library Join โดยส่งค่าพารามิเตอร์ตามที่ไลบรารีต้องการ คือ sSelect, sFrom, sWhere และ sCondition ไลบรารี MongoDB Library Join เมื่อได้รับค่าข้อมูลตามพารามิเตอร์ที่ระบุ จะนำค่าข้อมูลจากพารามิเตอร์ sCondition มาสร้าง query string condition ซึ่งเป็นเงื่อนไขในการสอบถามข้อมูล โดยสร้างให้อยู่ในรูปแบบคำสั่งสอบถามที่ใช้ในฐานข้อมูลมอโกดีบี กระบวนการต่อมาจะนำค่าข้อมูลจากพารามิเตอร์ sSelect, sFrom และ sWhere พร้อมกับ query string condition มาสร้างเป็น query string MapReduce

ขั้นตอนที่ 2 ไลบรารี MongoDB Library Join ส่งคำสั่ง query string condition_B ไปประมวลผลที่ฐานข้อมูลมอโกดีบี หลังจากประมวลผลคำสั่งเสร็จจะมีการสร้างคอลเลกชันชั่วคราวของคอลเลกชัน B เกิดขึ้น

ขั้นตอนที่ 3 ส่ง query string condition และ query string MapReduce ไปประมวลผลที่ฐานข้อมูลมอโกดีบี โดย query string MapReduce จะเรียกใช้งานผ่านคอลเลกชัน A ที่มี query string condition_A เพื่อกรองเฉพาะข้อมูลที่ต้องการ ต่อมาฟังก์ชันแมพจะส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชันชั่วคราวของคอลเลกชัน B โดยภายในฟังก์ชันแมพจะส่งข้อมูลไปยังฟังก์ชันรีดิวซ์ในรูปแบบ (คีย์, แวลู) ซึ่งในที่นี้คีย์เป็นค่าที่ไลบรารีสร้างให้ส่วนค่าแวลูเป็นค่าที่เกิดจากข้อมูลของคอลเลกชัน A และคอลเลกชันชั่วคราวของคอลเลกชัน B ต่อมาฟังก์ชันรีดิวซ์จะทำการรวมข้อมูลที่มาจากคอลเลกชัน A และคอลเลกชันชั่วคราวของคอลเลกชัน B ให้ตรงกับชื่อฟิลด์ที่ต้องการ เมื่อการทำงานของแมพรีดิวซ์ทำงานเสร็จจะมีการสร้างคอลเลกชันเอาต์พุตขึ้นมา

ขั้นตอนที่ 4 ไลบรารี MongoDB Library Join จะส่งคำสั่งสอบถามข้อมูลไปที่คอลเลกชันเอาต์พุตตามค่าพารามิเตอร์ sSelect เพื่อส่งผลลัพธ์กลับไปยังระบบผู้ใช้งาน

บทที่ 4 ระเบียบวิธีการวิจัย

ในงานวิจัยนี้ผู้วิจัยได้ดำเนินการทดสอบวัดประสิทธิภาพการเชื่อมความสัมพันธ์ของคอลเลกชันในฐานข้อมูลมองโกดีบี ผู้วิจัยได้ดำเนินการพัฒนาโปรแกรมที่ใช้ในการทดลอง โดยแบ่งโปรแกรมเป็นสองรูปแบบคือ รูปแบบที่หนึ่งดำเนินการพัฒนาโปรแกรมให้เป็นแบบไลบรารีเพื่อใช้สำหรับเชื่อมความสัมพันธ์ของคอลเลกชันในฐานข้อมูลมองโกดีบี แล้วสร้างหน้าเว็บไซต์ที่รองรับค่าพารามิเตอร์ขึ้นมาเพื่อเรียกใช้ไลบรารีที่สร้างขึ้นกับรูปแบบที่สองดำเนินการพัฒนาโปรแกรมโดยการสร้างหน้าเว็บไซต์ที่มีการเขียนโปรแกรมจัดการเชื่อมความสัมพันธ์ของคอลเลกชันของฐานข้อมูลมองโกดีบีไว้ในโดยตรง อีกทั้งผู้วิจัยได้สร้างกรณีทดสอบ การสอบถามข้อมูลที่มีเงื่อนไขแตกต่างกันจำนวนสามเงื่อนไขเพื่อใช้วัดประสิทธิภาพการเชื่อมความสัมพันธ์ของคอลเลกชันในฐานข้อมูลมองโกดีบีตามรูปแบบที่นำเสนอในบทที่ 3 ดังหัวข้อที่ 3.2 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง หัวข้อที่ 3.3 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์และหัวข้อที่ 3.4 รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว

4.1 วิธีการทดลองและการประเมินผล

ในการทดลองนั้น ผู้วิจัยได้ดำเนินการใช้เครื่องมือในการสร้างไลบรารีสำหรับเชื่อมความสัมพันธ์ของคอลเลกชันของฐานข้อมูลมองโกดีบี และสร้างหน้าเว็บไซต์ที่รองรับค่าพารามิเตอร์เพื่อเรียกใช้งานไลบรารีที่สร้างขึ้น ดังนี้ ใช้โปรแกรม Microsoft Visual Studio 2012 โดยใช้ภาษา C# ในการพัฒนาไลบรารีและหน้าเว็บไซต์ ใช้ฐานข้อมูลมองโกดีบี เวอร์ชัน 2.2.6 ในการจัดเก็บข้อมูลประกอบไปด้วยคอลเลกชันสองคอลเลกชัน คือคอลเลกชันที่หนึ่งเป็นคอลเลกชันสำหรับเก็บข้อมูลนักเรียน (student) กับคอลเลกชันที่สองเป็นคอลเลกชันสำหรับเก็บข้อมูลอาจารย์ที่ปรึกษา (advisor) โดยทั้งสองคอลเลกชันนั้นมีโครงสร้างข้อมูลดังแสดงในรูปที่ 4.1 ใช้โปรแกรม MongoVUE เวอร์ชัน 1.5.3 ซึ่งเป็นโปรแกรมที่มีอินเตอร์เฟซสำหรับจัดการฐานข้อมูลมองโกดีบี ใช้โปรแกรม WAPT 8.1 ในการจำลองสถานการณ์ทำรายการทดสอบเพื่อวัดค่าเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชันตามกรณีทดสอบที่ได้กำหนด โดยการทดลองครั้งนี้ได้ใช้โน้ตบุ๊ก CPU : Intel Core i5 1.8 GHz, RAM 8 GB, OS : Windows 8 จำลองเป็นเว็บเซิร์ฟเวอร์ ใช้พีซี CPU : Intel Core i3 3.1 GHz, RAM 4 GB, OS : Windows 7 จำลองเป็นดาต้าเบสเวิร์ฟเวอร์

คอลเลคชันนักเรียน			คอลเลคชันอาจารย์ที่ปรึกษา		
Field	Type	Index	Field	Type	Index
_id	ObjectId	Yes	_id	ObjectId	Yes
std_id	Int64	Yes	advisor_id	String	Yes
std_ename	String		efirst_name	String	
std_esurname	String		elast_name	String	
major_id	String		adv_id_new	String	
advisor_id	String	Yes	dept_id_new	String	
campus_id	String		adv_status	Int32	Yes
std_sex	String		major_id_new	String	
std_status	String	Yes			
degree	Int32				
extra_status	String				
admit_proj	String				

รูปที่ 4.1 โครงสร้างของคอลเลคชันนักเรียนกับอาจารย์ที่ปรึกษา

จากรูปที่ 4.1 แสดงโครงสร้างของคอลเลคชันนักเรียนกับคอลเลคชันอาจารย์ที่ปรึกษา โดยคอลเลคชันนักเรียนประกอบด้วย 14 ฟیلด์ มีฟیلด์ที่เป็นอินเด็กซ์คือ _id, std_id, advisor และ std_status คอลเลคชันอาจารย์ที่ปรึกษาประกอบด้วย 8 ฟیلด์ มีฟیلด์ที่เป็นอินเด็กซ์คือ _id, advisor_id และ adv_status

ผู้วิจัยได้จำลองสถานการณ์สอบถามข้อมูล โดยใช้ฟیلด์ที่เป็นเงื่อนไขให้เหมือนกันจำนวนสี่เงื่อนไขต่างกันตรงที่ค่าของฟیلด์ที่ทำให้เกิดจำนวนข้อมูลของคอลเลคชันต่างกัน โดยทดลองกับรูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันของฐานข้อมูลมอโกดีบีจำนวนสามรูปแบบคือ

รูปแบบที่หนึ่ง รูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการเขียนโปรแกรมโดยตรง เป็นวิธีการที่ผู้พัฒนาระบบเขียนโปรแกรมเพื่อจัดการเชื่อมความสัมพันธ์ภายในระบบด้วยตนเอง ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.2 โดยรูปแบบนี้ใช้ชื่อเรียกว่า Direct join

รูปแบบที่สอง รูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการใช้แมพรีดิวซ์ เป็นวิธีการเขียนโปรแกรมที่นำวิธีการเขียนโปรแกรมแบบแมพรีดิวซ์มาพัฒนาให้เป็นไลบรารีเพื่อประยุกต์ใช้เชื่อมความสัมพันธ์ของคอลเลคชัน ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.3 โดยรูปแบบนี้ใช้ชื่อเรียกว่า MR join

รูปแบบที่สาม รูปแบบการเชื่อมความสัมพันธ์ของคอลเลคชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลคชันชั่วคราว เป็นวิธีการเขียนโปรแกรมที่นำวิธีการเขียนโปรแกรมแบบแมพรีดิวซ์มาพัฒนาให้เป็นไลบรารีโดยภายในไลบรารีมีกระบวนการสร้างคอลเลคชันชั่วคราวของคอลเลคชันที่ไม่

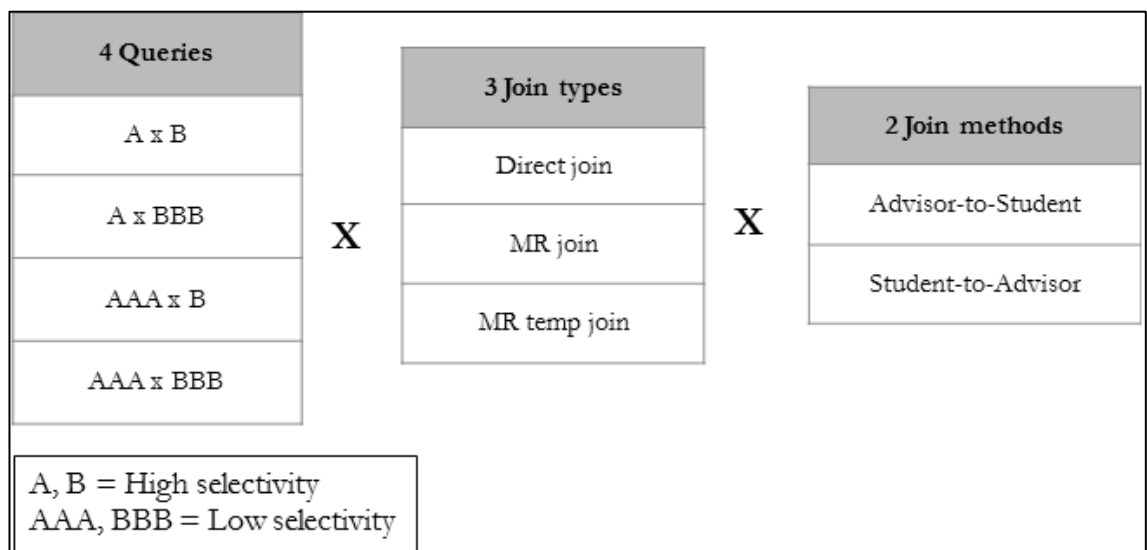
ถูกเรียกโดยแมพรีดิวซ์เพื่อใช้เชื่อมความสัมพันธ์จากคอลเลกชันที่ถูกเรียกโดยแมพรีดิวซ์ ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.4 โดยรูปแบบนี้ใช้ชื่อเรียกว่า MR temp join

โดยในการทดลองเชื่อมความสัมพันธ์ของคอลเลกชันแต่ละรูปแบบจะมีวิธีเชื่อมความสัมพันธ์ของคอลเลกชันสองวิธีคือ

วิธีที่หนึ่ง การเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B

วิธีที่สอง การเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน B ไปคอลเลกชัน A

โดยในการทดลองนี้คอลเลกชัน A หมายถึงคอลเลกชันนักเรียน และคอลเลกชัน B หมายถึงคอลเลกชันอาจารย์ที่ปรึกษา



รูปที่ 4.2 สถานการณ์ในการทดลองเพื่อสอบถามข้อมูล

จากรูปที่ 4.2 แสดงสถานการณ์ที่ใช้ในการทดลองเพื่อสอบถามข้อมูล โดยคิวรีที่ใช้ในการทดลองจะมีจำนวนสี่คิวรี ซึ่งลักษณะคิวรีจะมีสองแบบคือ high selectivity หมายถึงเงื่อนไขนั้นสามารถกรองข้อมูลได้มากทำให้เหลือข้อมูลน้อย ส่วน low selectivity หมายถึงเงื่อนไขนั้นสามารถกรองข้อมูลได้น้อยทำให้เหลือข้อมูลมาก แต่ละคิวรีจะทดลองกับรูปแบบการเชื่อมคอลเลกชันสามรูปแบบ และแต่ละรูปแบบจะมีวิธีการเชื่อมคอลเลกชันสองวิธี

ตารางที่ 4.1 แสดงจำนวนรายการข้อมูลของแต่ละเงื่อนไขที่ใช้ในการทดสอบการทำงานของไลบรารี

คิวรี	เงื่อนไข	จำนวนดอควิวเมนต์ของคอลเลกชัน	
		student (76,520)	advisor (6,118)
1	student.extra_status='N',student.admit_proj=52, student.major_id='Q03', advisor.dept_id_new='KQ04'	136 (0.18 %)	179 (2.93 %)
2	student.extra_status='N',student.admit_proj=52, student.major_id='KQ03', advisor.adv_status=3	136 (0.18 %)	3,558 (58.16 %)
3	student.extra_status='N', advisor.dept_id_new='KQ04'	47,684 (62.32 %)	179 (2.93 %)

จากตารางที่ 4.1 ประกอบด้วยสามคิวรี ซึ่งแต่ละคิวรีจะใช้ฟิลด์ที่เป็นเงื่อนไขแตกต่างกันเพื่อทำให้เกิดจำนวนรายการของคอลเลกชันต่างกัน โดยเงื่อนไขของ Query 1, Query 2 และ Query 3 จะมีลักษณะข้อมูลของคอลเลกชัน student กับคอลเลกชัน advisor เป็นแบบ (high selectivity , high selectivity), (high selectivity , low selectivity) และ (low selectivity , high selectivity) ซึ่งการทดลองในตารางที่ 4.1 นี้เพื่อแสดงให้เห็นว่าไลบรารีที่สร้างขึ้นสามารถรองรับค่าเงื่อนไขได้หลายค่า

ตารางที่ 4.2 แสดงจำนวนรายการข้อมูลของแต่ละเงื่อนไขที่ใช้ในการทดลอง

คิวรี	เงื่อนไข	จำนวนดอควิวเมนต์ของคอลเลกชัน	
		student (76,520)	advisor (6,118)
1	student.std_status='A', advisor.adv_status=6	1,188 (1.55 %)	348 (5.69 %)
2	student.std_status='A', advisor.adv_status=3	1,188 (1.55 %)	3,558 (58.16 %)
3	student.std_status='C', advisor.adv_status=6	36,463 (47.65 %)	348 (5.69 %)
4	student.std_status='C', advisor.adv_status=3	36,463 (47.65 %)	3,558 (58.16 %)

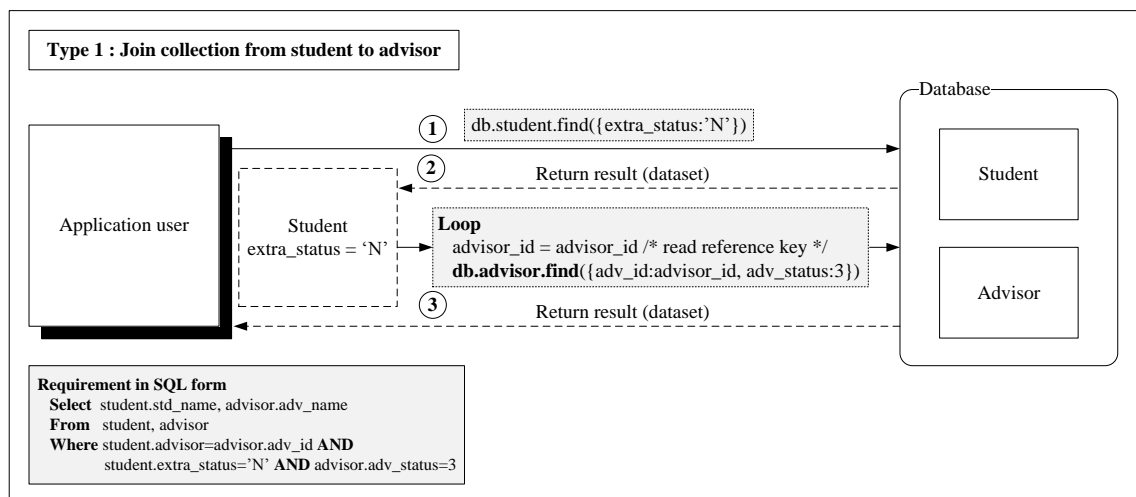
จากตารางที่ 4.2 ประกอบด้วยสี่คิวรี ซึ่งแต่ละคิวรีจะใช้ฟิลด์ที่เป็นเงื่อนไขเหมือนเดิมต่างกันตรงที่ค่าของฟิลด์ที่ทำให้เกิดจำนวนรายการของคอลเลกชันต่างกัน โดยเงื่อนไขของ Query 1, Query 2, Query 3 และ Query 4 จะมีลักษณะข้อมูลของคอลเลกชัน student กับคอลเลกชัน advisor เป็นแบบ

(high selectivity , high selectivity), (high selectivity , low selectivity) , (low selectivity , high selectivity) และ (low selectivity , low selectivity) ตามลำดับ ซึ่งการทดลองในกรณีนี้ 2 นี้เพื่อวัดค่าประสิทธิภาพการทำรายการเชื่อมคอลเลกชันของแต่ละรูปแบบที่นำเสนอในงานวิจัยนี้

ในการจำลองสถานการณ์เพื่อเชื่อมความสัมพันธ์ของคอลเลกชันในแต่ละรูปแบบสามารถอธิบายวิธีการทดลองได้ดังนี้

1.จำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมผู้ใช้งานกับฐานข้อมูลมอโกดีบีเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง (วิธี Direct join) โดยมีการจำลองหน้าเว็บไซต์ที่ภายในมีการเขียนโปรแกรมเชื่อมความสัมพันธ์ของคอลเลกชันไว้แล้ว มีส่วนประกอบคือ

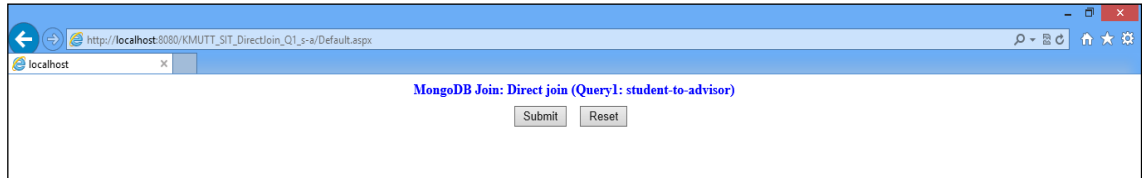
- หน้าเว็บไซต์โปรแกรมจำลองการเชื่อมความสัมพันธ์ของคอลเลกชันแบบ Direct join



รูปที่ 4.3 รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join

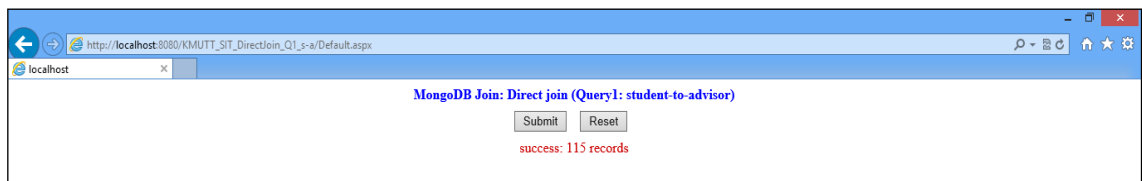
ในรูปที่ 4.3 แสดงรายละเอียดของการจำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมของผู้ใช้งานกับฐานข้อมูลมอโกดีบี เพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง Direct join ด้วยวิธีการเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B จากรูปถ้าต้องการเชื่อมความสัมพันธ์ระหว่างคอลเลกชัน Student และคอลเลกชัน Advisor จะเริ่มด้วยการส่งคำสั่งสอบถามข้อมูลพร้อมระบุเงื่อนไขที่ต้องการไปยังคอลเลกชัน Student ดังหมายเลข 1 ซึ่งผลลัพธ์จากคำสั่งสอบถามข้อมูลจะอยู่ในรูปของชุดข้อมูล (dataset) ดังหมายเลข 2 หลังจากนั้นภายในโปรแกรมจะทำการอ่านค่าเรฟเฟอร์เรนซ์คีย์ ซึ่งตามรูปคือค่า advisor_id เพื่อนำไปสอบถามในคอลเลกชัน B ร่วมกับค่าเงื่อนไขของคอลเลกชัน B ดังหมายเลข 3 เมื่อฐานข้อมูลประมวลผลคำสั่ง

สอบถามข้อมูลเสร็จจะส่งผลลัพธ์จากคอลเลกชัน B กลับไปยังระบบผู้ใช้งาน ซึ่งเป็นไปตามกระบวนการในบทที่ 3 หัวข้อที่ 3.2



รูปที่ 4.4 หน้าเว็บไซต์จำลองการเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join

ในรูปที่ 4.4 แสดงหน้าเว็บไซต์จำลองการเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join ซึ่งภายในเว็บไซต์มีการเขียน โปรแกรมเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันไว้แล้ว โดยวิธีทดสอบเชื่อมคอลเลกชันทำได้โดยการกดปุ่ม Submit เพื่อเริ่มการเชื่อมความสัมพันธ์ของคอลเลกชัน

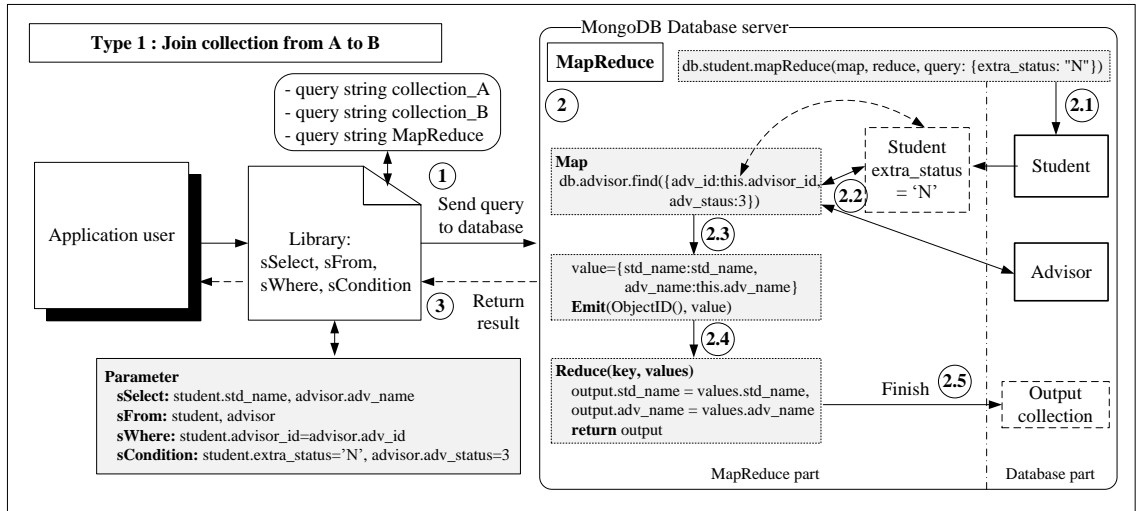


รูปที่ 4.5 แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่หนึ่ง Direct join เสร็จสิ้น

ในรูปที่ 4.5 แสดงข้อความและจำนวนข้อมูลที่เชื่อมความสัมพันธ์ได้เมื่อกระบวนการเชื่อมความสัมพันธ์ของคอลเลกชันดำเนินการเสร็จสิ้น

2. จำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมผู้ใช้งานกับฐานข้อมูลมอโกดีบีเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ (วิธี MR join) โดยดำเนินการพัฒนาโปรแกรมให้เป็นไลบรารีเพื่อใช้สำหรับเชื่อมคอลเลกชัน มีการสร้างหน้าเว็บไซต์จำลองการรับค่าพารามิเตอร์เพื่อส่งให้กับไลบรารีที่สร้างขึ้น มีส่วนประกอบดังต่อไปนี้

- หน้าเว็บไซต์จำลองการเชื่อมความสัมพันธ์ของคอลเลกชัน (MR join)
- โปรแกรม MongoVUE สำหรับดูผลจากการเชื่อมคอลเลกชัน



รูปที่ 4.6 รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join

ในรูปที่ 4.6 แสดงรายละเอียดของการจำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมของผู้ใช้งานกับฐานข้อมูลมอองโกดีบี เพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ MR join โดยมีการพัฒนาโปรแกรมให้เป็นไลบรารีที่รองรับค่าพารามิเตอร์เพื่อใช้เชื่อมคอลเลกชันด้วยวิธีการเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B จากรูปถ้าต้องการเชื่อมความสัมพันธ์ระหว่างคอลเลกชัน Student และคอลเลกชัน Advisor จะเริ่มจากระบบผู้ใช้งานส่งค่าพารามิเตอร์ไปยังไลบรารีที่พัฒนาขึ้น ภายในไลบรารีจะทำการสร้างชุดคำสั่งเพื่อส่งไปยังฐานข้อมูลมอองโกดีบีดังกล่าวหมายเลข 1 หลังจากนั้นกระบวนการเชื่อมคอลเลกชันโดยแมพรีดิวซ์จะเริ่มทำงานผ่านการเรียกจากคอลเลกชัน Student พร้อมระบุเงื่อนไขที่ต้องการ ดังหมายเลข 2.1 หลังจากนั้นฟังก์ชันแมพจะส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชัน Advisor โดยใช้ค่าเรฟเฟอร์เรนซ์คีย์ในที่นี้คือค่า advisor_id ซึ่งมาจากพารามิเตอร์ sWhere ของคอลเลกชัน Student รวมกับเงื่อนไขของคอลเลกชัน Advisor ในการค้นหาข้อมูลที่ต้องการ ดังหมายเลข 2.2 เพื่อให้ข้อมูลอยู่ในรูปแบบของฟังก์ชันแมพคือ (คีย์, แวลู) ซึ่งค่าคีย์คือค่า ObjectID ที่ถูกสร้างโดยฐานข้อมูลมอองโกดีบี ส่วนค่าแวลูคือค่าของข้อมูลที่มาจากคอลเลกชัน Student และคอลเลกชัน Advisor ดังหมายเลข 2.3 หลังจากนั้นฟังก์ชันรีดิวซ์จะทำการอ่านค่าข้อมูลเพื่อกำหนดให้ตรงตามชื่อฟิลด์ที่ต้องการ ดังหมายเลข 2.4 หลังจากกระบวนการเชื่อมคอลเลกชันโดยแมพรีดิวซ์ทำงานเสร็จจะสร้างเป็นคอลเลกชันเอาต์พุตขึ้นมา ดังหมายเลข 2.5 จากนั้นไลบรารีจะทำการส่งคำสั่งสำหรับสอบถามข้อมูลที่มาจากพารามิเตอร์ sSelect ไปยังคอลเลกชันเอาต์พุตเพื่อส่งข้อมูลกลับไปยังระบบผู้ใช้งาน ซึ่งเป็นไปตามกระบวนการในบทที่ 3 หัวข้อที่ 3.3

The screenshot shows a web browser window with the URL `http://localhost:8080/KMUTT_SIT_MRJoin/Default.aspx`. The page title is "MongoDB Join: MR join". The form contains four text input fields labeled "Select Parameter:", "From Parameter:", "Where Parameter:", and "Condition Parameter:". At the bottom of the form are two buttons: "Submit" and "Reset".

รูปที่ 4.7 หน้าเว็บไซท์จำลองการรับพารามิเตอร์เพื่อเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join

ในรูปที่ 4.7 แสดงหน้าเว็บไซท์จำลองการรับพารามิเตอร์เพื่อส่งค่าให้กับไลบรารีสำหรับเชื่อมคอลเลกชัน ซึ่งประกอบไปด้วยฟิลด์รับค่าทั้งหมดสี่ฟิลด์ได้แก่ ฟิลด์ที่หนึ่ง Select Parameter คือชื่อฟิลด์ที่ต้องการให้แสดงข้อมูลจากคอลเลกชัน ฟิลด์ที่สอง From Parameter คือชื่อของคอลเลกชันที่ต้องการเชื่อมความสัมพันธ์ ฟิลด์ที่สาม Where Parameter คือชื่อฟิลด์ที่เป็นคีย์ที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชันและฟิลด์ที่สี่ Condition Parameter คือชื่อฟิลด์ที่เป็นเงื่อนไขในการแสดงข้อมูล โดยวิธีทดสอบเชื่อมคอลเลกชันทำได้โดยการใส่ข้อมูลครบทั้งสี่ฟิลด์ ตัวอย่างในการใส่ค่าพารามิเตอร์ต่างๆ ดังนี้

พารามิเตอร์ Select Parameter ค่าที่ใส่คือ

kstudent.std_id (ชื่อฟิลด์เก็บข้อมูลรหัสนักเรียน),
 kstudent.std_ename (ชื่อฟิลด์เก็บข้อมูลชื่อนักเรียน),
 kstudent.std_esurname (ชื่อฟิลด์เก็บข้อมูลนามสกุลนักเรียน),
 kadvisor.first_name (ชื่อฟิลด์เก็บข้อมูลชื่ออาจารย์),
 kadvisor.lastname (ชื่อฟิลด์เก็บข้อมูลนามสกุลอาจารย์)

พารามิเตอร์ From Parameter ค่าที่ใส่คือ

kstudent (ชื่อคอลเลกชันนักเรียน),
 kadvisor (ชื่อคอลเลกชันอาจารย์ที่ปรึกษา)

พารามิเตอร์ Where Parameter ค่าที่ใส่คือ

kstudent.advisor_id= kadvisor. advisor_id (ระบุฟิลด์ที่เป็นคีย์ที่ใช้เชื่อมความสัมพันธ์กันของคอลเลกชัน)

พารามิเตอร์ Condition Parameter ค่าที่ใส่คือ

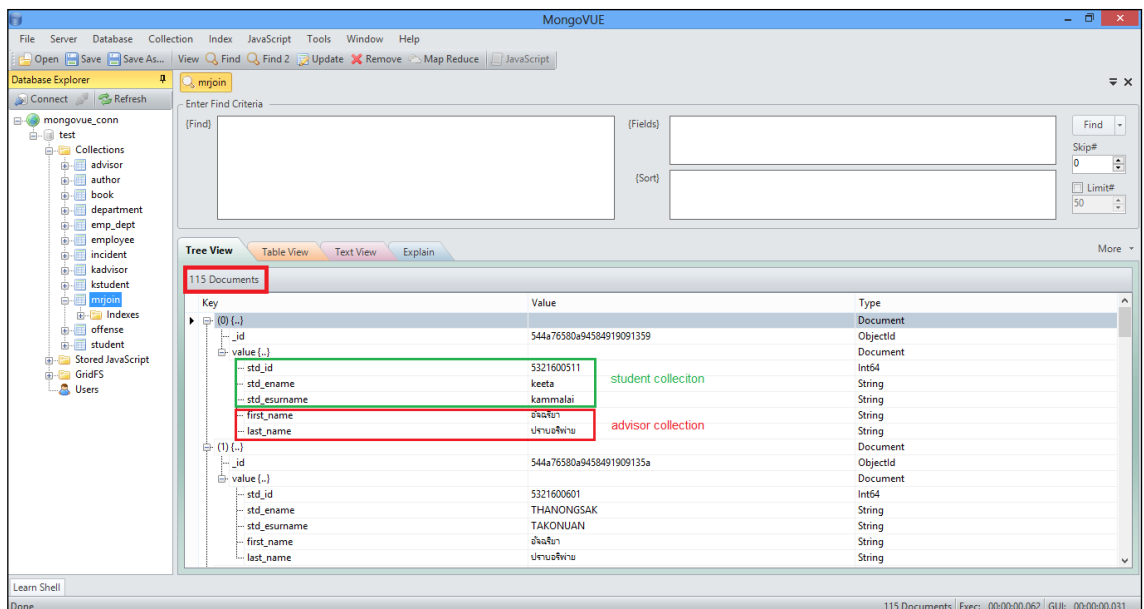
kstudent.extra_status='N', kstudent.admit_proj=52, kstudent.major_id='Q03',
 kadvisor.dept_new='KQ04' (เงื่อนไขในการสอบถามข้อมูล)

เมื่อดำเนินการใส่ค่าพารามิเตอร์ต่างๆแล้ว ดำเนินการกดปุ่ม Submit เพื่อเริ่มส่งคำสั่งเพื่อดำเนินการสอบถามข้อมูลที่มีการเชื่อมความสัมพันธ์ของคอลเลกชัน



รูปที่ 4.8 แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join เสร็จสิ้น

ในรูปที่ 4.8 แสดงข้อความและจำนวนข้อมูลที่เชื่อมความสัมพันธ์ได้เมื่อกระบวนการเชื่อมความสัมพันธ์ของคอลเลกชันดำเนินการเสร็จสิ้น

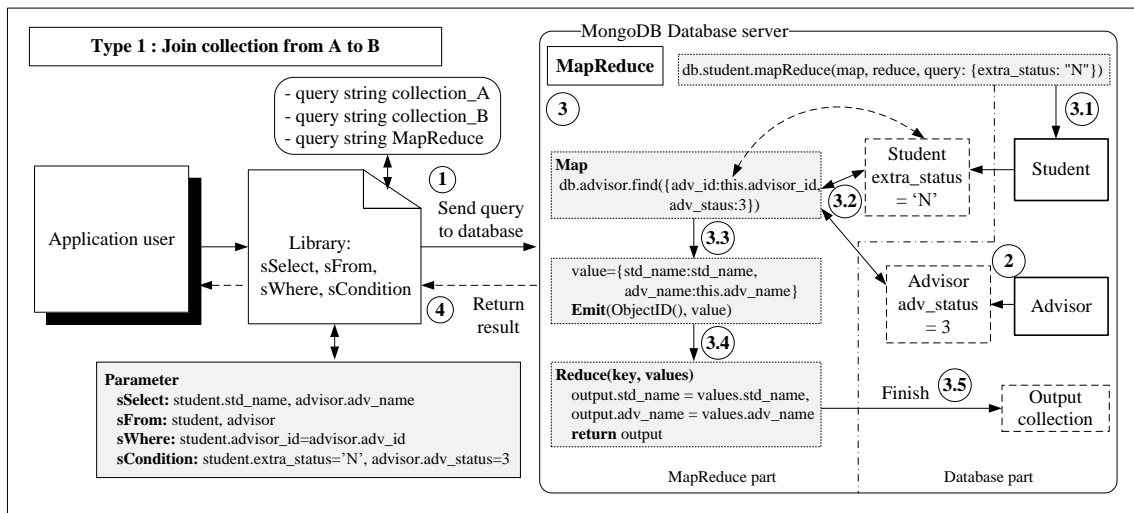


รูปที่ 4.9 แสดงผลการเชื่อมคอลเลกชันตามรูปแบบที่สอง MR join

ในรูปที่ 4.9 แสดงผลลัพธ์ที่ได้จากการสอบถามข้อมูลที่มีการเชื่อมความสัมพันธ์ของคอลเลกชันด้วยวิธีการใช้แมพรีดิวซ์ MR join ผลลัพธ์ข้อมูลที่ได้นั้นมาจากคอลเลกชันนักเรียนและคอลเลกชันอาจารย์ที่ปรึกษา ซึ่งผลลัพธ์จะเป็นค่าฟิลด์ข้อมูลที่แสดงตามค่าที่ระบุในพารามิเตอร์ Select Parameter สามารถดูผลลัพธ์ได้จากโปรแกรม MongoVUE

3.จำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมผู้ใช้งานกับฐานข้อมูลมอโกดีบีเพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว (วิธี MR temp join) โดยดำเนินการพัฒนาโปรแกรมให้เป็นไลบรารีซึ่งภายในไลบรารีมีกระบวนการสร้างคอลเลกชันชั่วคราวของคอลเลกชันที่ไม่ถูกเรียกโดยแมพรีดิวซ์เพื่อใช้เชื่อมความสัมพันธ์จากคอลเลกชันที่ถูกเรียกโดยแมพรีดิวซ์ มีส่วนประกอบดังต่อไปนี้

- หน้าเว็บไซต์จำลองการเชื่อมความสัมพันธ์ของคอลเลกชัน (MR temp join)
- โปรแกรม MongoVUE สำหรับดูผลจากการเชื่อมคอลเลกชัน



รูปที่ 4.10 รายละเอียดการจำลองสถานการณ์การเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join

ในรูปที่ 4.10 แสดงรายละเอียดของการจำลองสถานการณ์การติดต่อสื่อสารระหว่างโปรแกรมของผู้ใช้งานกับฐานข้อมูลมอโกดีบี เพื่อเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ MR join โดยมีการพัฒนาโปรแกรมให้เป็นไลบรารีที่รองรับค่าพารามิเตอร์เพื่อใช้เชื่อมคอลเลกชันด้วยวิธีการเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B จากรูปถ้าต้องการเชื่อมความสัมพันธ์ระหว่างคอลเลกชัน Student และคอลเลกชัน Advisor จะเริ่มจากระบบผู้ใช้งานส่งค่าพารามิเตอร์ไปยังไลบรารีที่พัฒนาขึ้น ภายในไลบรารีจะทำการสร้างชุดคำสั่งเพื่อส่งไปยังฐานข้อมูลมอโกดีบีดังหมายเลข 1 ภายในไลบรารีจะมีกระบวนการสร้างคอลเลกชันชั่วคราวที่มีข้อมูลเฉพาะที่ต้องการของคอลเลกชัน Advisor ขึ้นมาก่อน ดังหมายเลขที่ 2 หลังจากนั้นกระบวนการเชื่อมคอลเลกชันโดยแมพรีดิวซ์จะเริ่มทำงานผ่านการเรียกจากคอลเลกชัน Student พร้อมระบุเงื่อนไขที่ต้องการ ดังหมายเลข 3.1 หลังจากนั้นฟังก์ชันแมพจะส่งคำสั่งสอบถามข้อมูลไปยังคอลเลกชันชั่วคราวของคอลเลกชัน Advisor ที่สร้างขึ้นโดยใช้ค่าเรฟเฟอร์เรนซ์คีย์ ในที่นี้คือค่า advisor_id ซึ่งมาจากพารามิเตอร์ sWhere ของคอลเลกชัน Student ในการค้นหาข้อมูลที่ต้องการ ดังหมายเลข 3.2 เพื่อให้ข้อมูลอยู่ใน

รูปแบบของฟังก์ชันแมพคือ (คีย์,แวลู) ซึ่งค่าคีย์คือค่า ObjectID ที่ถูกสร้างโดยฐานข้อมูลมองโกดีบี ส่วนค่าแวลูคือค่าของข้อมูลที่มาจากคอลเลกชัน Student และคอลเลกชันชั่วคราวของคอลเลกชัน Advisor ดังหมายเลข 3.3 หลังจากนั้นฟังก์ชันรีดิวซ์จะทำการอ่านค่าข้อมูลเพื่อกำหนดให้ตรงตามชื่อฟิลด์ที่ต้องการ ดังหมายเลข 3.4 หลังจากกระบวนการเชื่อมคอลเลกชัน โดยแมพรีดิวซ์ทำงานเสร็จจะสร้างเป็นคอลเลกชันเอาต์พุตขึ้นมา ดังหมายเลข 3.5 จากนั้นไลบรารีจะทำการส่งคำสั่งสำหรับสอบถามข้อมูลที่มาจากพารามิเตอร์ sSelect ไปยังคอลเลกชันเอาต์พุตเพื่อส่งข้อมูลกลับไปยังระบบผู้ใช้งาน ซึ่งเป็นไปตามกระบวนการในบทที่ 3 หัวข้อที่ 3.4

รูปที่ 4.11 หน้าเว็บไซต์จำลองการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join

ในรูปที่ 4.11 แสดงหน้าเว็บไซต์จำลองการรับพารามิเตอร์เพื่อส่งค่าให้กับไลบรารีสำหรับเชื่อมคอลเลกชัน ซึ่งประกอบไปด้วยฟิลด์รับค่าทั้งหมดสี่ฟิลด์ได้แก่ ฟิลด์ที่หนึ่ง Select Parameter คือชื่อฟิลด์ที่ต้องการแสดงข้อมูลจากคอลเลกชัน ฟิลด์ที่สอง From Parameter คือชื่อของคอลเลกชันที่ต้องการเชื่อมความสัมพันธ์ ฟิลด์ที่สาม Where Parameter คือชื่อฟิลด์ที่เป็นคีย์ที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชันและฟิลด์ที่สี่ Condition Parameter คือชื่อฟิลด์ที่เป็นเงื่อนไขในการแสดงข้อมูล โดยวิธีทดสอบเชื่อมคอลเลกชันทำได้โดยการใส่ข้อมูลครบทั้งสี่ฟิลด์ ตัวอย่างในการใส่ค่าพารามิเตอร์ต่างๆ ดังนี้

พารามิเตอร์ Select Parameter ค่าที่ใส่คือ

- kstudent.std_id (ชื่อฟิลด์เก็บข้อมูลรหัสนักเรียน),
- kstudent.std_ename (ชื่อฟิลด์เก็บข้อมูลชื่อนักเรียน),
- kstudent.std_esurname (ชื่อฟิลด์เก็บข้อมูลนามสกุลนักเรียน),
- kadvisor.first_name (ชื่อฟิลด์เก็บข้อมูลชื่ออาจารย์),
- kadvisor.lastname (ชื่อฟิลด์เก็บข้อมูลนามสกุลอาจารย์)

พารามิเตอร์ From Parameter ค่าที่ใส่คือ

- kstudent (ชื่อคอลเลกชันนักเรียน),
- kadvisor (ชื่อคอลเลกชันอาจารย์ที่ปรึกษา)

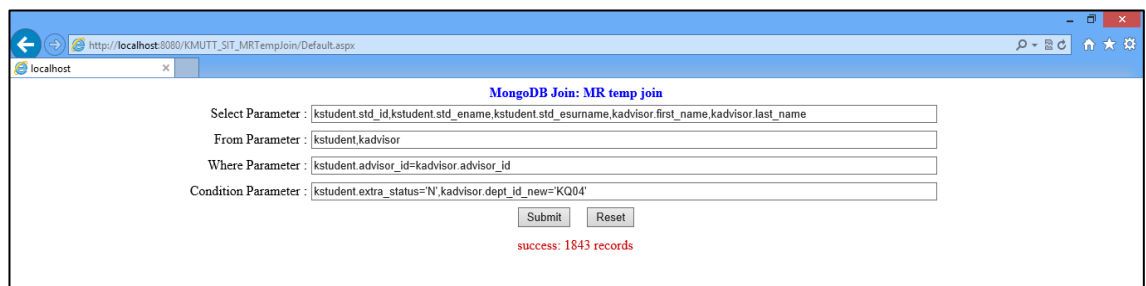
พารามิเตอร์ Where Parameter ค่าที่ใส่คือ

`kstudent.advisor_id= kadvisor.advisor_id` (ระบุฟิลด์ที่เป็นคีย์ที่ใช้เชื่อม
ความสัมพันธ์กันของคอลเลกชัน)

พารามิเตอร์ Condition Parameter ค่าที่ใส่คือ

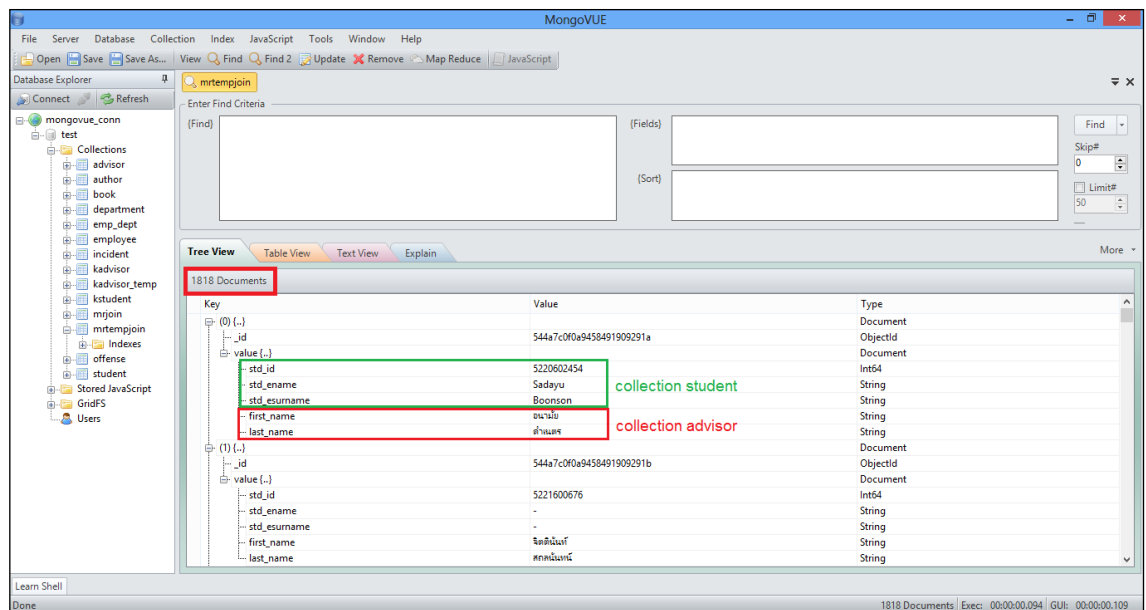
`kstudent.extra_status='N',kadvisor.dept_new='KQ04'` (เงื่อนไขในการสอบถาม
ข้อมูล)

เมื่อดำเนินการใส่ค่าพารามิเตอร์ต่างๆแล้ว ดำเนินการกดปุ่ม Submit เพื่อเริ่มการเชื่อมความสัมพันธ์
ของคอลเลกชัน



รูปที่ 4.12 แสดงข้อความเมื่อกระบวนการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join เสร็จสิ้น

ในรูปที่ 4.12 แสดงข้อความและจำนวนข้อมูลที่เชื่อมความสัมพันธ์ได้เมื่อกระบวนการเชื่อม
ความสัมพันธ์ของคอลเลกชันดำเนินการเสร็จสิ้น



รูปที่ 4.13 แสดงผลการเชื่อมคอลเลกชันตามรูปแบบที่สาม MR temp join

ในรูปที่ 4.13 แสดงผลการเชื่อมความสัมพันธ์ของคอลเลกชันซึ่งมีข้อมูลที่มาจากคอลเลกชันนักเรียน และคอลเลกชันอาจารย์ที่ปรึกษาโดยดูจากโปรแกรม MongoVUE

ในการทดลองเพื่อวัดประสิทธิภาพค่าเวลาการทำรายการนั้นจะทดลองกับรูปแบบการเชื่อมความสัมพันธ์ทั้งสามรูปแบบได้แก่

รูปแบบที่หนึ่ง Direct join

รูปแบบที่สอง MR join

รูปแบบที่สาม MR temp join

โดยแบ่งตามวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันดังนี้

- การวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันด้วยวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B
- การวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันด้วยวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน B ไปคอลเลกชัน A

โดยมีรายละเอียดการทดลองแต่ละวิธีในหัวข้อถัดไป

4.2 การวัดประสิทธิภาพค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไป

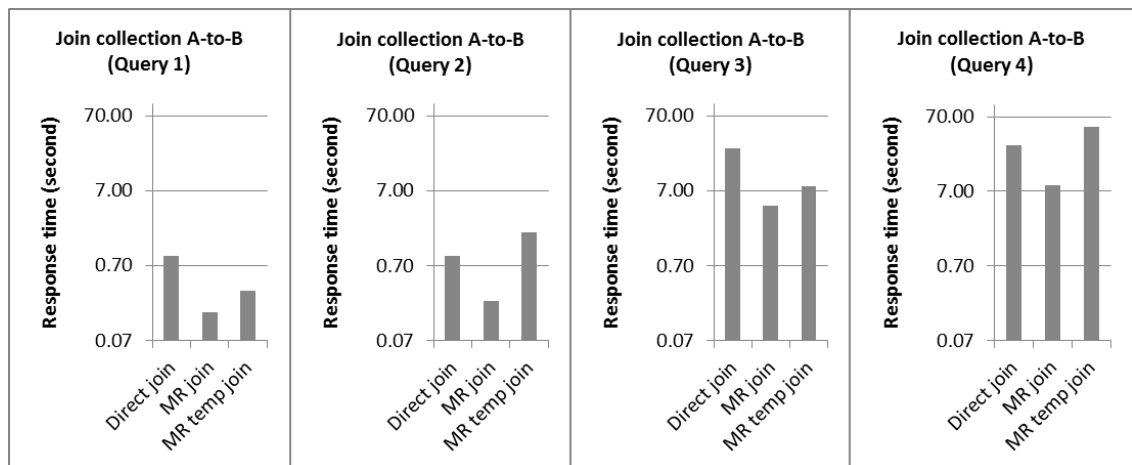
คอลเลกชัน B

การวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันด้วยวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B นั้น จะทำการทดลองกับรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันที่ได้นำเสนอทั้งสามรูปแบบได้แก่ รูปแบบที่หนึ่ง Direct join รูปแบบที่สอง MR join และรูปแบบที่สาม MR temp join เพื่อวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชันแต่ละรูปแบบ ในการทดลองวัดค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันนั้น จะทดลองตามตารางที่ 4.2 ซึ่งจะดำเนินการทดลองเงื่อนไขละสามรอบของแต่ละวิธีเพื่อวัดประสิทธิภาพค่าเวลาเฉลี่ยในการเชื่อมคอลเลกชัน โดยในการทดลองนี้กำหนดให้คอลเลกชัน A หมายถึงคอลเลกชันนักเรียน (student) และคอลเลกชัน B หมายถึงคอลเลกชันอาจารย์ที่ปรึกษา (advisor)

ตารางที่ 4.3 ค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไปคอลเลกชัน B

คิวรี	จำนวนดอควิวเมนต์		จำนวนดอควิวเมนต์ในผลลัพธ์ของคิวรี	เวลาที่ใช้ในการประมวลผล (วินาที)		
	student (76,520)	advisor (6,118)		Direct join	MR join	MR temp join
Query 1	1,188 (1.55 %)	348 (5.69 %)	11	0.94	0.17	0.32
Query 2	1,188 (1.55 %)	3,558 (58.16 %)	538	0.95	0.24	1.96
Query 3	36,463 (47.65 %)	348 (5.69 %)	62	26.00	4.41	8.05
Query 4	36,463 (47.65 %)	3,558 (58.16 %)	34,437	28.90	8.43	50.80

จากตารางที่ 4.3 เป็นค่าเวลาเฉลี่ยที่ใช้ในการทดลองเชื่อมความสัมพันธ์ของคอลเลกชัน จากคอลเลกชัน student ไปคอลเลกชัน advisor ทั้งสามรูปแบบ ได้แก่ รูปแบบที่หนึ่ง Direct join รูปแบบที่สอง MR join และรูปแบบที่สาม MR temp join โดยแต่ละรูปแบบจะทดลองทั้งหมดสี่คิวรี ในแต่ละคิวรีจะทดลองสามรอบและนำค่าเวลาที่วัดได้แสดงในตารางที่ 4.3 มาสร้างเป็นกราฟได้ดังรูปที่ 4.14



รูปที่ 4.14 กราฟแสดงค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน A ไปคอลเลกชัน B

ในรูปที่ 4.14 แสดงกราฟเปรียบเทียบค่าเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชัน จากคอลเลกชัน A (student) ไปคอลเลกชัน B (advisor) ของรูปแบบ Direct join รูปแบบ MR join และรูปแบบ

MR temp join พบว่ากราฟจาก Query 1 และ Query 3 จะให้ผลในลักษณะแบบเดียวกันคือรูปแบบ Direct join จะใช้เวลาในการเชื่อมคอลเลกชันนานสุดเนื่องจากกระบวนการเชื่อมคอลเลกชันต้องสูญเสียเวลาในการติดต่อระหว่างระบบของผู้ใช้งานกับฐานข้อมูลอยู่ตลอดเวลาจนกว่าจะสิ้นสุดการทำงาน ในขณะที่รูปแบบ MR join และรูปแบบ MR temp join จะไม่ต้องสูญเสียเวลาในส่วนนี้เพราะกระบวนการเชื่อมคอลเลกชันเกิดขึ้นที่ฝั่งฐานข้อมูลเป็นหลักโดยเฉพาะรูปแบบ MR join ที่ใช้เวลาน้อยกว่าทุกรูปแบบในทุกคิวรี ส่วนกราฟจาก Query 2 และ Query 4 ที่รูปแบบ MR temp join ใช้เวลาในการเชื่อมคอลเลกชันนานกว่ารูปแบบ Direct join เนื่องจากเงื่อนไขของ Query 2 และ Query 4 กรองข้อมูลของคอลเลกชัน B ได้น้อยจึงทำให้มีปริมาณข้อมูลมาก ทำให้การสร้างคอลเลกชันชั่วคราวของคอลเลกชัน B ต้องสูญเสียเวลาไปมาก นอกจากนี้เมื่อมีการสร้างคอลเลกชันชั่วคราวแล้วคอลเลกชันชั่วคราวที่สร้างขึ้นจะไม่มีอินเด็กซ์ทำให้กระบวนการเชื่อมคอลเลกชันต้องใช้หลักการสแกน เป็นผลให้การเชื่อมคอลเลกชันของ Query 2 และ Query 4 ด้วยรูปแบบ MR temp join ทำงานได้ช้า แต่ถ้าคอลเลกชันชั่วคราวที่สร้างขึ้นมีข้อมูลน้อยดังเช่น Query 1 และ Query 3 ก็จะทำให้การทำงานเร็วขึ้น เพราะการสแกนข้อมูลเพียงเล็กน้อยย่อมทำงานได้เร็วกว่าการใช้อินเด็กซ์กับข้อมูลที่มีปริมาณมาก

4.3 การวัดประสิทธิภาพค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไป

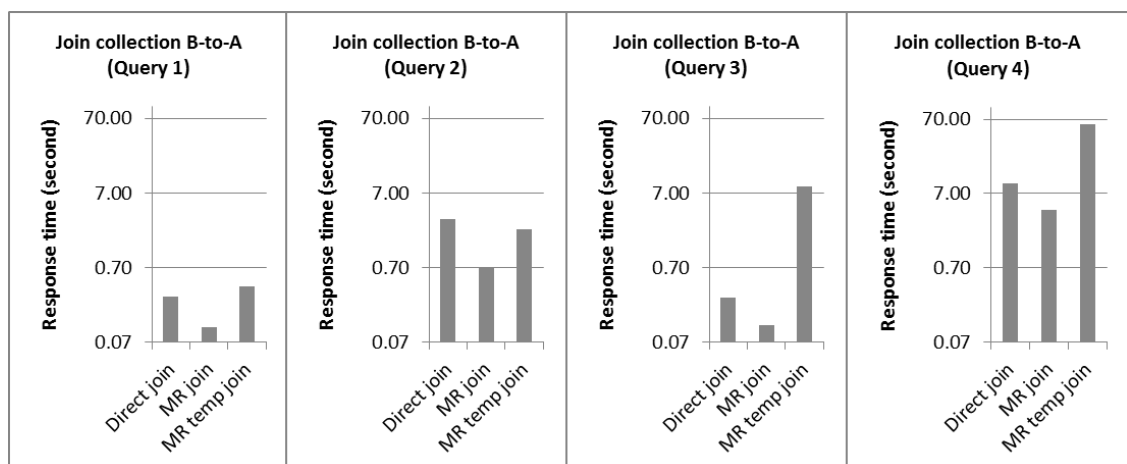
คอลเลกชัน A

การวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมคอลเลกด้วยวิธีการเชื่อมความสัมพันธ์ของคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A นั้นจะทำการทดลองกับรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันที่ได้นำเสนอทั้งสามรูปแบบได้แก่ รูปแบบที่หนึ่ง Direct join รูปแบบที่สอง MR join และรูปแบบที่สาม MR temp join เพื่อวัดประสิทธิภาพค่าเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชันแต่ละรูปแบบ ในการทดลองวัดค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันจะทดลองตามตารางที่ 4.2 ซึ่งจะดำเนินการทดลองเงื่อนไขและสามรอบของแต่ละวิธีเพื่อวัดประสิทธิภาพค่าเวลาเฉลี่ยในการเชื่อมคอลเลกชัน โดยในการทดลองนี้กำหนดให้คอลเลกชัน A หมายถึงคอลเลกชันนักเรียน (student) และคอลเลกชัน B หมายถึงคอลเลกชันอาจารย์ที่ปรึกษา (advisor)

ตารางที่ 4.4 ค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A

คิวรี	จำนวนดอควิวเมนต์		จำนวนดอควิวเมนต์ในผลลัพธ์ของคิวรี	เวลาที่ใช้ในการประมวลผล (วินาที)		
	student 76,520	advisor 6,118		Direct join	MR join	MR temp join
Query 1	1,188 (1.55 %)	348 (5.69 %)	11	0.29	0.11	0.39
Query 2	1,188 (1.55 %)	3,558 (58.16 %)	538	3.13	0.72	2.25
Query 3	36,463 (47.65 %)	348 (5.69 %)	62	0.28	0.12	8.41
Query 4	36,463 (47.65 %)	3,558 (58.16 %)	34,437	9.49	4.26	59.60

จากตารางที่ 4.4 เป็นค่าเวลาเฉลี่ยที่ใช้ในการทดลองเชื่อมความสัมพันธ์ของคอลเลกชัน จากคอลเลกชัน student ไปคอลเลกชัน advisor ทั้งสามรูปแบบได้แก่ รูปแบบที่หนึ่ง Direct join รูปแบบที่สอง MR join และรูปแบบที่สาม MR temp join โดยแต่ละรูปแบบจะทดลองทั้งหมดสี่คิวรีในแต่ละคิวรีจะทดลองสามรอบและนำค่าเวลาที่วัดได้แสดงในตาราง 4.4 นำมาสร้างเป็นกราฟได้ดังรูปที่ 4.15



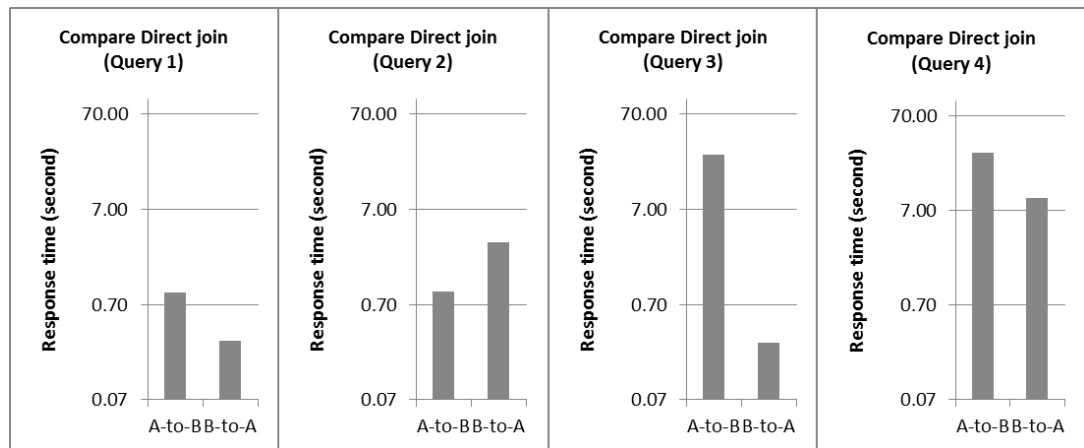
รูปที่ 4.15 กราฟแสดงค่าเวลาการเชื่อมคอลเลกชัน จากคอลเลกชัน B ไปคอลเลกชัน A

ในรูปที่ 4.15 แสดงกราฟเปรียบเทียบค่าเวลาที่ใช้ในการเชื่อมความสัมพันธ์ของคอลเลกชัน จากคอลเลกชัน B (advisor) ไปคอลเลกชัน A (student) ของรูปแบบ Direct join รูปแบบ MR join และรูปแบบ

MR temp join พบว่ารูปแบบ MR join ใช้เวลาในการเชื่อมคอลเลคชันน้อยกว่าทุกรูปแบบในทุกคิวรี เนื่องจากกระบวนการเชื่อมคอลเลคชันเกิดขึ้นที่ฝั่งฐานข้อมูลเป็นหลักทำให้ไม่ต้องสูญเสียเวลาในการโอนถ่ายข้อมูลระหว่างระบบผู้ใช้กับฐานข้อมูล ส่วนรูปแบบ MR temp join โดยเฉพาะ Query 3 และ Query 4 นั้น แม้จะมีกระบวนการเชื่อมคอลเลคชันที่ฝั่งฐานข้อมูลเป็นหลักแต่กลับใช้เวลาในการเชื่อมคอลเลคชันค่อนข้างนานกว่ารูปแบบ Direct join พอสมควรเนื่องจากเงื่อนไขการกรองข้อมูลของคอลเลคชัน A ทำได้น้อย ทำให้มีปริมาณข้อมูลที่ต้องสร้างคอลเลคชันชั่วคราวมากประกอบกับคอลเลคชันชั่วคราวที่สร้างขึ้นไม่มีอินเด็กซ์ทำให้กระบวนการเชื่อมคอลเลคชันต้องใช้หลักการสแกน ซึ่งทั้งสองสาเหตุนี้ส่งผลให้การเชื่อมคอลเลคชันทำได้ช้าลง

4.4 การเปรียบเทียบค่าเวลาการทำรายการ

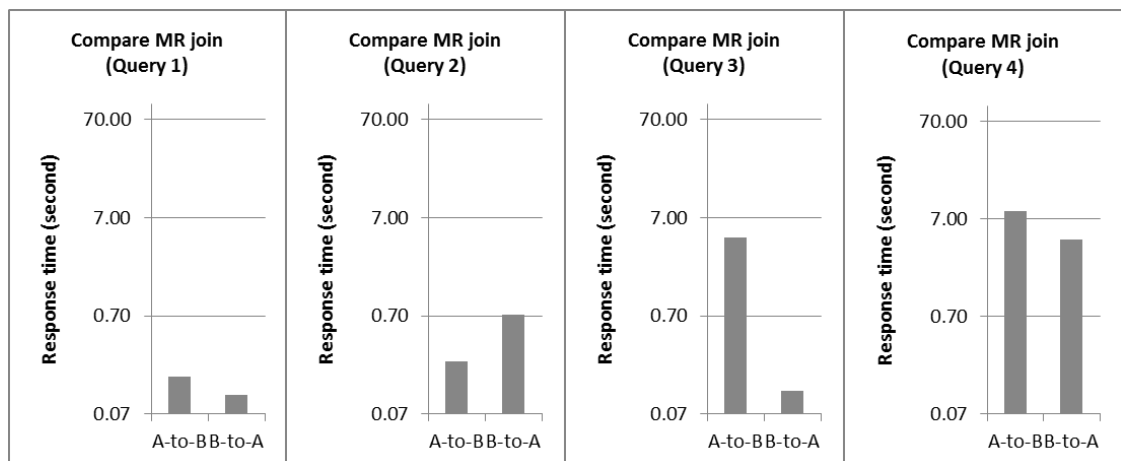
จากการวัดค่าเวลาในการทำรายการเชื่อมความสัมพันธ์ของคอลเลคชันทั้งสามรูปแบบได้แก่ รูปแบบที่หนึ่ง Direct join รูปแบบที่สอง MR join และรูปแบบที่สาม MR temp join กับวิธีการเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน A ไปคอลเลคชัน B ดังหัวข้อที่ 4.2 และวิธีการเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน B ไปคอลเลคชัน A ดังหัวข้อที่ 4.3 สามารถนำค่าเวลาการทำรายการที่วัดได้มาสร้างเป็นกราฟเปรียบเทียบดังนี้



รูปที่ 4.16 กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีการเชื่อมคอลเลคชันรูปแบบ Direct join

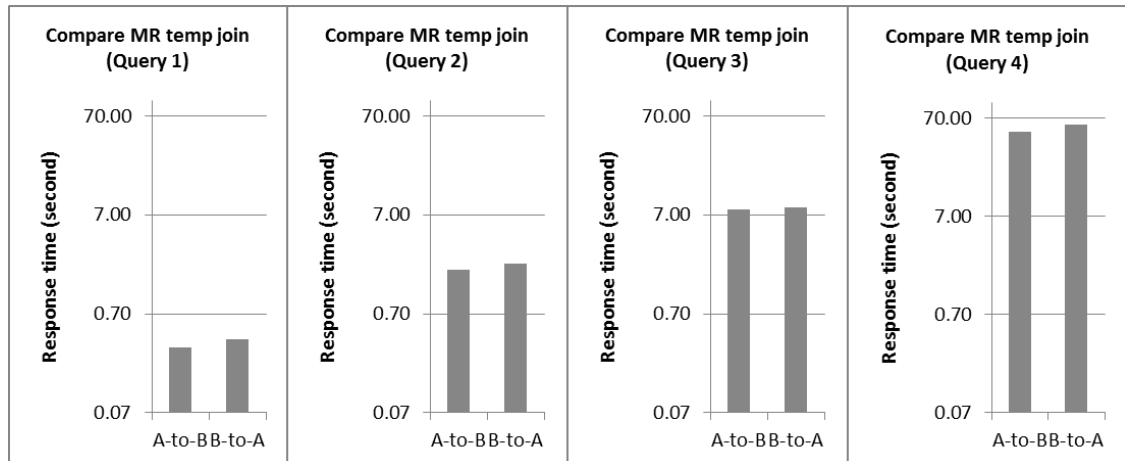
จากรูปที่ 4.16 แสดงกราฟเปรียบเทียบค่าเวลาของวิธีการเชื่อมความสัมพันธ์ของคอลเลคชันรูปแบบ Direct join ระหว่างการเชื่อมคอลเลคชัน A (student) ไปคอลเลคชัน B (advisor) กับการเชื่อมคอลเลคชัน B ไปคอลเลคชัน A พบว่ากราฟจาก Query 1, Query 3 และ Query 4 จะให้ผลในลักษณะแบบเดียวกันคือการเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน B ไปคอลเลคชัน A จะใช้เวลาน้อยกว่าการเชื่อมความสัมพันธ์ของคอลเลคชันจากคอลเลคชัน A ไปคอลเลคชัน B เนื่องจากเงื่อนไขของ

Query 1, Query 3 และ Query 4 กรองข้อมูลของคอลเลกชัน A ได้มากจึงทำให้ปริมาณข้อมูลที่จะต้องเชื่อมความสัมพันธ์มีน้อยส่งผลให้กระบวนการเชื่อมความสัมพันธ์ใช้เวลาสั้นลง ในขณะที่ Query 2 การเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน B ไปคอลเลกชัน A จะใช้เวลามากกว่าการเชื่อมความสัมพันธ์ของคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B เนื่องจากเงื่อนไขของ Query 2 กรองข้อมูลของคอลเลกชัน A ได้น้อยจึงทำให้ปริมาณข้อมูลที่จะต้องเชื่อมความสัมพันธ์มีมากส่งผลให้กระบวนการเชื่อมความสัมพันธ์ต้องใช้เวลาเพิ่มขึ้น



รูปที่ 4.17 กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีการเชื่อมคอลเลกชันรูปแบบ MR join

จากรูปที่ 4.17 แสดงกราฟเปรียบเทียบค่าเวลาของวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันรูปแบบ MR join ระหว่างการเชื่อมคอลเลกชัน A (student) ไปคอลเลกชัน B (advisor) กับ การเชื่อมคอลเลกชัน B ไปคอลเลกชัน A พบว่าผลการทดลองมีลักษณะไปในทิศทางเดียวกันกับรูปแบบ Direct join ที่ได้กล่าวไว้ตามกราฟในรูปที่ 4.16 ซึ่งแสดงให้เห็นว่าการเชื่อมคอลเลกชันที่กรองข้อมูลได้มากไปหาคอลเลกชันที่กรองข้อมูลได้น้อยจะเชื่อมคอลเลกชันได้เร็วกว่าการเชื่อมคอลเลกชันที่กรองข้อมูลได้น้อยไปหาคอลเลกชันที่กรองข้อมูลได้มาก



รูปที่ 4.18 กราฟแสดงการเปรียบเทียบค่าเวลาของวิธีเชื่อมคอลเลกชันรูปแบบ MR temp join

จากรูปที่ 4.18 แสดงกราฟเปรียบเทียบค่าเวลาของวิธีการเชื่อมความสัมพันธ์ของคอลเลกชันรูปแบบ MR temp join ระหว่างการเชื่อมคอลเลกชัน A (student) ไปคอลเลกชัน B (advisor) กับ การเชื่อมคอลเลกชัน B ไปคอลเลกชัน A พบว่าผลการทดลองไม่ได้ต่างกันมากนัก ค่อนข้างจะใกล้เคียงกันมาก ซึ่งแสดงให้เห็นว่าการเชื่อมคอลเลกชันด้วยรูปแบบ MR temp join นั้น ปริมาณข้อมูลที่เกิดจากการกรองด้วยเงื่อนไขไม่ได้มีนัยสำคัญต่อวิธีการเชื่อมคอลเลกชันไม่ว่าจะเป็นการเชื่อมคอลเลกชัน A ไปคอลเลกชัน B หรือการเชื่อมคอลเลกชัน B ไปคอลเลกชัน A เพราะท้ายที่สุดแล้วก็ต้องใช้เวลาไปกับการสร้างคอลเลกชันชั่วคราว

บทที่ 5

บทสรุป

ในงานวิจัยนี้ผู้วิจัยได้ดำเนินการศึกษาฐานข้อมูลโนเอสคิวแอลประเภท Document Data Model โดยฐานข้อมูลประเภท Document Data Model ที่ได้ทำการศึกษาคือฐานข้อมูลมองโกดีบีเน้นการศึกษาในเรื่องของรูปแบบการจัดเก็บข้อมูลและการนำมาใช้งาน โดยฐานข้อมูลมองโกดีบีนั้นจะมีความยืดหยุ่นในเรื่องของโครงสร้างการจัดเก็บข้อมูลสามารถเก็บข้อมูลซ้อนเป็นลำดับชั้นได้ ข้อมูลจะถูกเก็บอยู่ภายในคอลลัมน์และแต่ละคอลลัมน์จะถูกเก็บอยู่ภายในคอลเลกชัน ดังนั้นหากมีการจัดเก็บข้อมูลที่มีความสัมพันธ์กันอยู่คนละคอลเลกชันคล้ายกับการเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์ ถ้าต้องการสอบถามข้อมูลที่มีความสัมพันธ์กันที่จัดเก็บในฐานข้อมูลมองโกดีบีเพื่อนำข้อมูลไปใช้งาน ผู้ที่ทำการสอบถามข้อมูลจะต้องส่งคำสั่งไปค้นหาที่ละคอลเลกชันเพื่อแสดงผลลัพธ์เนื่องจากฐานข้อมูลมองโกดีบีไม่มีชุดคำสั่งที่ใช้ค้นหาข้อมูลที่มีความสัมพันธ์กันแต่อยู่ต่างคอลเลกชัน

งานวิจัยนี้จึงได้นำเสนอวิธีการเชื่อมความสัมพันธ์ของข้อมูลที่อยู่ต่างคอลเลกชันโดยใช้แมพรีดิวซ์ซึ่งแมพรีดิวซ์นั้นเป็นวิธีการเขียนโปรแกรมเพื่อประมวลผลข้อมูลขนาดใหญ่ โดยแมพรีดิวซ์นั้นจะประกอบไปด้วยสองขั้นตอนคือขั้นตอนแมพและขั้นตอนรีดิวซ์ซึ่งงานวิจัยนี้ได้ทดลองสร้างรูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันไว้สามรูปแบบคือรูปแบบที่หนึ่ง การเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง เป็นรูปแบบที่ผู้พัฒนาระบบต้องมีการเขียนโปรแกรมเพื่อจัดการเชื่อมคอลเลกชันด้วยตนเอง ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.2 รูปแบบที่สอง การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ เป็นการเขียนโปรแกรมที่นำวิธีของแมพรีดิวซ์มาประยุกต์แล้วพัฒนาให้เป็นไลบรารีสำหรับเชื่อมความสัมพันธ์ของคอลเลกชัน ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.3 และรูปแบบที่สาม การเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิวซ์ร่วมกับการสร้างคอลเลกชันชั่วคราว เป็นการเขียนโปรแกรมที่นำวิธีของแมพรีดิวซ์มาประยุกต์แล้วพัฒนาให้เป็นไลบรารีโดยภายในไลบรารีมีกระบวนการสร้างคอลเลกชันชั่วคราวขึ้นมาก่อนที่จะเริ่มกระบวนการเชื่อมคอลเลกชันด้วยแมพรีดิวซ์เพื่อลดจำนวนข้อมูลที่จะเชื่อมความสัมพันธ์ ตามกระบวนการในบทที่ 3 หัวข้อที่ 3.4

5.1 บทสรุปของงานวิจัย

จากงานวิจัยการใช้แมพรีดิคชันเพื่อเชื่อมความสัมพันธ์ของข้อมูลที่อยู่ต่างคอลเลกชันของฐานข้อมูลมองโกดีบีสรุปได้ดังต่อไปนี้

1) รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการเขียนโปรแกรมโดยตรง ใช้เวลาในการเชื่อมคอลเลกชันค่อนข้างนานเมื่อเทียบกับรูปแบบอื่น โดยเฉพาะถ้าเงื่อนไขกรองข้อมูลได้น้อยจะทำให้มีปริมาณข้อมูลที่ใช้เชื่อมคอลเลกชันมาก ส่งผลให้สูญเสียเวลาในการโอนถ่ายข้อมูลระหว่างระบบผู้ใช้งานกับฐานข้อมูลเนื่องจากรูปแบบนี้กระบวนการเชื่อมคอลเลกชันจะเกิดที่ระบบของผู้ใช้งานเป็นหลัก

2) รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิคชัน ใช้เวลาในการเชื่อมคอลเลกชันค่อนข้างเร็วกว่ารูปแบบอื่นไม่ว่าเงื่อนไขนั้นจะกรองข้อมูลได้มากหรือน้อยก็ตามเนื่องจากรูปแบบนี้กระบวนการเชื่อมคอลเลกชันส่วนใหญ่จะอยู่ที่ฐานข้อมูลเป็นหลักทำให้ลดการสูญเสียเวลาในการโอนถ่ายข้อมูล

3) รูปแบบการเชื่อมความสัมพันธ์ของคอลเลกชันจากการใช้แมพรีดิคชันร่วมกับการสร้างคอลเลกชันชั่วคราว ใช้เวลาในการเชื่อมคอลเลกชันค่อนข้างนานเมื่อเทียบกับรูปแบบอื่น โดยเฉพาะถ้าเงื่อนไขกรองข้อมูลได้น้อยจะทำให้มีปริมาณข้อมูลที่ต้องสร้างคอลเลกชันชั่วคราวมาก นอกจากนี้เมื่อมีการสร้างคอลเลกชันชั่วคราวแล้วคอลเลกชันชั่วคราวที่สร้างขึ้นจะไม่มีอินเด็กซ์ทำให้กระบวนการเชื่อมคอลเลกชันต้องใช้หลักการสแกน เป็นผลให้การเชื่อมคอลเลกชันทำงานได้ช้าลง แต่ถ้าคอลเลกชันชั่วคราวที่สร้างขึ้นมีข้อมูลน้อยก็จะทำให้การทำงานเร็วขึ้นเพราะการสแกนข้อมูลเพียงเล็กน้อยย่อมทำงานได้เร็วกว่าการใช้อินเด็กซ์กับข้อมูลที่มีปริมาณมาก

4) ในการพิจารณาเลือกวิธีเชื่อมความสัมพันธ์ของคอลเลกชันระหว่างคอลเลกชัน A ไปคอลเลกชัน B กับคอลเลกชัน B ไปคอลเลกชัน A นั้น ควรเลือกเชื่อมคอลเลกชันจากคอลเลกชันที่เงื่อนไขสามารถกรองข้อมูลได้มากไปหาคอลเลกชันที่เงื่อนไขกรองข้อมูลได้น้อย เพื่อให้มีปริมาณข้อมูลที่ใช้ในการเชื่อมคอลเลกชันมีน้อย ซึ่งจะส่งผลให้กระบวนการเชื่อมคอลเลกชันทำได้เร็วขึ้น

5) ทำให้ได้รูปแบบเพื่อใช้เชื่อมความสัมพันธ์ของข้อมูล เป็นการลดข้อจำกัดที่ฐานข้อมูลโนเอสคิวแอลไม่มีชุดคำสั่งที่ใช้ในการเชื่อมความสัมพันธ์ของข้อมูลที่แยกเก็บคนละส่วน นอกจากนี้ยังช่วยให้ผู้พัฒนาระบบไม่จำเป็นต้องเขียนโปรแกรมเพื่อส่งคำสั่งเพื่อไปสอบถามข้อมูลยังฐานข้อมูลมองโกดีบีโดยตรง ลดการเขียนโปรแกรมที่ซับซ้อนในการวนชุดคำสั่งเพื่อไปสอบถามข้อมูลมองโกดีบีโดยตรง เพียงแค่ส่งพารามิเตอร์ตามที่ระบบต้องการเท่านั้น ผู้ที่สนใจสามารถนำแนวคิดนี้ไปประยุกต์ใช้กับฐานข้อมูลโนเอสคิวแอลรูปแบบอื่นได้

5.2 แนวทางที่พัฒนาในอนาคตและข้อเสนอแนะ

เนื่องจากโปรแกรมที่พัฒนาขึ้นนั้นผู้ใช้งานยังคงต้องระบุวิธีการเชื่อมคอลเลกชันว่าจะเชื่อมจากคอลเลกชัน A ไปคอลเลกชัน B หรือคอลเลกชัน B ไปคอลเลกชัน A ซึ่งสำหรับงานในอนาคตของงานวิจัยนี้ผู้ที่สนใจสามารถนำแนวคิดนี้ไปต่อยอดได้ เช่น ปรับให้ระบบมีโหมดการทำงานสองโหมดคือ โหมดการเชื่อมคอลเลกชันจากผู้ใช้เป็นคนเลือกกับโหมดการเชื่อมคอลเลกชันจากระบบเลือกให้ ซึ่งคล้ายคลึงกับหลักการ Query Processing ในการวิเคราะห์ว่าควรจะเชื่อมคอลเลกชันจากคอลเลกชัน A ไปคอลเลกชัน B หรือจากคอลเลกชัน B ไปคอลเลกชัน A เพื่อเป็นการเพิ่มทางเลือกให้กับผู้พัฒนาระบบเพราะผู้พัฒนาระบบหรือผู้ใช้งานอาจไม่ทราบว่าปริมาณข้อมูลที่จัดเก็บในแต่ละฟิลด์นั้นมีค่าของข้อมูลที่แตกต่างกันมากน้อยเพียงใดเพราะฟิลด์ของข้อมูลเหล่านั้นมีผลต่อเงื่อนไขที่ใช้ในการค้นหาข้อมูล ซึ่งส่งผลโดยตรงต่อการเชื่อมความสัมพันธ์ของคอลเลกชันตามรูปแบบที่ได้นำเสนอไว้

ในส่วนของข้อเสนอแนะนั้น ในปัจจุบันฐานข้อมูลโนเอสคิวแอลเริ่มเป็นที่นิยมมากขึ้นเพราะสามารถรองรับการจัดเก็บข้อมูลที่มีโครงสร้างและชนิดของข้อมูลที่หลากหลายได้ ผู้ที่สนใจสามารถนำแนวคิดนี้ไปประยุกต์ใช้กับงานของตนได้และอาจมีการพัฒนาประสิทธิภาพในเรื่องของการเชื่อมข้อมูลให้มีประสิทธิภาพมากขึ้น

เอกสารอ้างอิง

1. Cattell, R., 2010, “Scalable SQL and NoSQL data stores”, **ACM SIGMOD Record**, Vol. 39, No. 4, pp. 12-27.
2. Wang, G. and Tang, J., 2012, “The NoSQL Principles and Basic Application of Cassandra Model”, **2012 International Conference on Computer Science & Service System**, August 2012, China, pp. 1332-1335
3. Han, J., E, H., Le, G. and Du, J., 2011, “Survey on NoSQL Database”, **The 6th of International Conference on Pervasive Computing and Applications**, South Africa, pp. 363-366.
4. MongoDB Inc, **Who Uses MongoDB?** [Online], Available : <http://www.mongodb.com/who-uses-mongodb/>. [2014, September 21].
5. Tiwari, S., 2011, **Professional NoSQL**, 1st ed., John Wiley & Sons Inc, Canada, p. 4.
6. Sadalage, P.J. and Fowler, M., 2013, **NoSQL distilled : a brief guide to the emerging world of polyglot**, 2nd ed., Pearson Education Inc, United States of America, pp. 9-28.
7. Islam, R., 2011, **PHP and MongoDB Web Development Beginner's Guide**, 1st ed., Packt Publishing, Birmingham, pp. 8-12.
8. Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E. and Abramov, J., 2011, “Security Issues in NoSQL Databases”, **The 10th of IEEE International Conference on Trust, Security and Privacy in Computing and Communications**, Changsha, pp. 541-547.
9. MongoDB Org, **SQL to MongoDB Mapping Chart** [Online], Available : <http://docs.mongodb.org/manual/reference/sql-comparison>. [2013, September 2].

10. Shroff, G., 2010, **Enterprise Cloud Computing Technology Architecture Application**, 1st ed., Cambridge University Press, United Kingdom, pp. 131-142.
11. MongoDB Org, **Map-Reduce** [Online], Available :
<http://docs.mongodb.org/manual/core/map-reduce/>. [2015, May 4].
12. Calil, A. and Mello, R.S., 2012, "SimpleSQL: A Relational Layer for SimpleDB", **ADBIS'12: Proceedings of the 16th East European conference on Advances in Databases and Information Systems**, September 2012, Germany, pp. 99-110.
13. Wei, Z., Pierre, G. and Chi, C., 2012, "Scalable Join Queries in Cloud Data Stores", **CCGRID '12 Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2012)**, USA, pp. 547-555.
14. Lui, L., Yin, J. and Gao, L., 2013, "Efficient Social Network Data Query Processing on Mapreduce", **HotPlanet'13**, 16 August 2013, China, pp. 27-32.
15. Myung, J., Yeon, J. and Lee, S., 2010, "SPARQL Basic Graph Pattern Processing with Iterative MapReduce", **MDAC '10**, 26 April 2012, USA, p. 6.
16. Karam, S., 2013, **Linked Data RDF and SPARQL Part 1** [Online], Available :
<http://feed.askmaclean.com/archives/linked-data-rdf-and-sparql-part-1.html>. [2015, May 10].
17. Dean, J. and Ghemawat, S., 2004, "MapReduce: Simplified Data Processing on Large Clusters", **OSDI**, 2004, pp. 1-13.
18. nosql-database.org, **List Of NoSQL Databases** [Online], Available :
<http://nosql-database.org/>. [2013, October 5].
19. Harris, S. and Seaborne, A., 2013, **SPARQL 1.1 Query Language** [Online], Available :
<http://www.w3.org/TR/sparql11-query/>. [2015, May 5].

20. Redis, **Introduction to Redis** [Online], Available : <http://redis.io/topics/introduction>.
[2015, June 7].
21. Basho, **Why Riak** [Online], Available : <http://docs.basho.com/riak/latest/theory/why-riak/>.
[2015, June 7].
22. FAL Labs, **Tokyo Cabinet: a modern implementation of DBM** [Online], Available :
<http://fallabs.com/tokyocabinet/>. [2015, June 7].
23. MongoDB Org, **Introduction to MongoDB** [Online], Available :
<http://www.mongodb.org/about/introduction/>. [2015, June 7].
24. Apache CouchDB, **Technical Overview** [Online], Available :
<http://docs.couchdb.org/en/latest/intro/overview.html>. [2015, June 7].
25. DataStax, **About Apache Cassandra** [Online], Available :
<http://docs.datastax.com/en/cassandra/2.0/cassandra/gettingStartedCassandraIntro.html>.
[2015, June 8].
26. IBM, **What is HBase** [Online], Available :
<http://www-01.ibm.com/software/data/infosphere/hadoop/hbase/>. [2015, June 8].
27. Neo Technology Inc, **Get Started** [Online], Available :
<http://neo4j.com/developer/get-started/>. [2015, June 8].
28. Twitter Inc, **Introducing FlockDB** [Online], Available :
<https://blog.twitter.com/2010/introducing-flockdb>. [2015, June 8].

ประวัติผู้วิจัย

ชื่อ – สกุล	นายนิรุทธ์ รวยรื่น
วัน เดือน ปีเกิด	10 มิถุนายน 2527
ประวัติการศึกษา	
ระดับมัธยมศึกษา	ประโยคมัธยมศึกษาตอนปลาย โรงเรียนรัตนราษฎร์บำรุง พ.ศ. 2545
ระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน พ.ศ. 2550
ระดับปริญญาโท	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี พ.ศ. 2557
ทุนการศึกษา หรือทุนวิจัย	ทุนการศึกษาระดับบัณฑิตศึกษา สาขาเทคโนโลยีสารสนเทศ ปังบประมาณ 2556
ประวัติการทำงาน	- โปรแกรมเมอร์ บริษัท ธนาคารชนชาติ จำกัด พ.ศ. 2551-2554 - นักวิชาการคอมพิวเตอร์ กองบริหารวิชาการและนิตินิต มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขต กำแพงแสน พ.ศ. 2554-2558
ผลงานที่ได้รับการตีพิมพ์	นิรุทธ์ รวยรื่น และ เกรียงไกร ปอแก้ว, 2557, “การใช้แมพรีดิคส์ เชื่อมคอลเลกชันของฐานข้อมูลโนเอสคิวแอลบนมองโกดีบี”, การ ประชุมวิชาการเสนอผลงานวิจัยระดับบัณฑิตศึกษา, ครั้งที่ 15, 28 มีนาคม 2557, มหาวิทยาลัยขอนแก่น, หน้า 342-350.

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ข้อตกลงว่าด้วยการโอนลิขสิทธิ์วิทยานิพนธ์

วันที่...24...เดือน...มิถุนายน...พ.ศ...2558..

ข้าพเจ้า (นาย/นาง/นางสาว).....นิรุทธ์ รวยรื่น.....รหัสประจำตัว.....55440389.....
เป็นนักศึกษาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ระดับปริญญา โท เอก
หลักสูตร วิทยาศาสตร์มหาบัณฑิต สาขาวิชา เทคโนโลยีสารสนเทศ คณะ เทคโนโลยีสารสนเทศ
อยู่บ้านเลขที่.....42.....ตรอก/ซอย.....-.....ถนน.....-.....
ตำบล/แขวง.....ท่าผา.....อำเภอ/เขต.....บ้านโป่ง.....จังหวัด.....ราชบุรี.....
รหัสไปรษณีย์.....70110.....เป็น “ผู้โอน” ขอโอนลิขสิทธิ์วิทยานิพนธ์ให้กับ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี โดยมี รศ.ดร.นิพนธ์ เจริญกิจการ ตำแหน่ง คณบดี
คณะเทคโนโลยีสารสนเทศ เป็นตัวแทน “ผู้รับโอน” สิทธิในทรัพย์สินทางปัญญาและมีข้อตกลง
ดังนี้

1. ข้าพเจ้าได้จัดทำวิทยานิพนธ์เรื่อง...การใช้แมพริควิชเชื่อมคอลเลกชันของฐานข้อมูล
มองโกดีบี....
ซึ่งอยู่ในความควบคุมของ..... ผศ.ดร.เกรียงไกร ปอแก้ว
ตามมาตราพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 และถือว่าเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
2. ข้าพเจ้าตกลงโอนลิขสิทธิ์จากผลงานทั้งหมดที่เกิดขึ้นจากการสร้างสรรค์ของข้าพเจ้า
ในวิทยานิพนธ์ให้กับมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ตลอดอายุแห่งการคุ้มครอง
ลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 ตั้งแต่วันที่ได้รับอนุมัติโครงร่างวิทยานิพนธ์จาก
มหาวิทยาลัย
3. ในกรณีที่ข้าพเจ้าประสงค์จะนำวิทยานิพนธ์ไปใช้ในการเผยแพร่ในสื่อใด ๆ ก็ตาม
ข้าพเจ้าจะต้องระบุว่าวิทยานิพนธ์เป็นผลงานของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีทุก
ครั้งที่มีการเผยแพร่
4. ในกรณีที่ข้าพเจ้าประสงค์จะนำวิทยานิพนธ์ไปเผยแพร่หรืออนุญาตให้ผู้อื่นทำซ้ำหรือ
ดัดแปลงหรือเผยแพร่ต่อสาธารณชนหรือกระทำการอื่นใด ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537
โดยมีค่าตอบแทนในเชิงธุรกิจ ข้าพเจ้าจะกระทำได้เมื่อได้รับความยินยอมเป็นลายลักษณ์อักษรจาก
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีก่อน
5. ในกรณีที่ข้าพเจ้าประสงค์จะนำข้อมูลจากวิทยานิพนธ์ไปประดิษฐ์หรือพัฒนาต่อยอด
เป็นสิ่งประดิษฐ์หรืองานทรัพย์สินทางปัญญาประเภทอื่น ภายในระยะเวลา (10) ปีนับจากวันลงนาม
ในข้อตกลงฉบับนี้ ข้าพเจ้าจะกระทำได้เมื่อได้รับความยินยอมเป็นลายลักษณ์อักษรจากมหาวิทยาลัย
เทคโนโลยีพระจอมเกล้าธนบุรี และมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีมีสิทธิในทรัพย์สิน
ทางปัญญานั้น พร้อมกับได้รับชำระค่าตอบแทนการอนุญาตให้ใช้สิทธิดังกล่าว รวมถึงการจัดสรร

ผลประโยชน์อันพึงเกิดขึ้นจากส่วนใดส่วนหนึ่งหรือทั้งหมดของวิทยานิพนธ์ในอนาคต โดยให้
เป็นไปตามระเบียบสถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ว่าด้วยการบริหารผลประโยชน์อันเกิด
จากทรัพย์สินทางปัญญา พ.ศ. 2538

6. ในกรณีที่มีผลประโยชน์เกิดขึ้นจากวิทยานิพนธ์หรืองานทรัพย์สินทางปัญญาอื่นที่
ข้าพเจ้าทำขึ้น โดยมีมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีเป็นเจ้าของ ข้าพเจ้าจะมีสิทธิ์ได้รับ
การจัดสรรผลประโยชน์อันเกิดจากทรัพย์สินทางปัญญาดังกล่าวตามอัตราที่กำหนดไว้ในระเบียบ
สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ว่าด้วยการบริหารผลประโยชน์อันเกิดจากทรัพย์สินทาง
ปัญญา พ.ศ. 2538

ลงชื่อ.....^{ศิริดา ยุทธะนันท์}.....ผู้โอนลิขสิทธิ์

(นายนิรุทธ์ รวยรื่น)

ลงชื่อ.....^{รศ.ดร.นิพนธ์ เจริญกิจการ}.....ผู้รับโอนลิขสิทธิ์

(รศ.ดร.นิพนธ์ เจริญกิจการ)

ลงชื่อ.....^{ศิริดา ยุทธะนันท์}.....พยาน

(นางศิริดา ยุทธะนันท์)

ลงชื่อ.....^{ฮารัลด์ เชิงชาฎุเกษตร}.....พยาน

(นางสาวฮารัลด์ เชิงชาฎุเกษตร)