การศึกษาเปรียบเทียบประสิทธิภาพระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบใช้คีย์อ้างอิงสำหรับขอมูลขนาดใหญ่

A COMPARATIVE STUDY OF RELATIONAL DATABASE AND KEY – VALUE DATABASE FOR BIG DATA

วิทธวัช พวงสายใจ
WITTAWAT PUANGSAIJAI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2560
KMITL-2017-EN-M-230-139

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาเปรียบเทียบประสิทธิภาพระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบใช้คีย์อ้างอิงสำหรับข้อมูลขนาดใหญ่

A COMPARATIVE STUDY OF RELATIONAL DATABASE AND KEY – VALUE DATABASE FOR BIG DATA

T148682

วิทธวัช พวงสายใจ

WITTAWAT PUANGSAIJAI

ตรพม.
เลขทะเบียน 148682
วัน.เดือน.ปี 14 พ.ย. 2560

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

KMITL-2017-EN-M-230-139

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A COMPARATIVE STUDY OF RELATIONAL DATABASE AND KEY – VALUE DATABASE FOR BIG DATA

WITTAWAT PUANGSAIJAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2017

KMITL-2017-EN-M-230-139

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2017

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การศึกษาเปรียบเทียบประสิทธิภาพระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบใช้คีย์อ้างอิงสำหรับข้อมูลขนาดใหญ่

Thesis Title A Comparative Study Study of Relational Database and Key Value Database for Big Data

นักศึกษา นายวิทธวัช พวงสายใจ

รหัสประจำตัว 57601304

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.สุธีรา พันธุ์ธีรานุรักษ์

หมายเลขวิทยานิพนธ์ KMITL-2017-EN-M-230-139

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.เอื้อน	ปิ่นเงิน	
รศ.ดร.อรรถสิทธิ์	หล่าสกุล	
ผศ.ดร.พนารัตน์	เชิญถนอมวงศ์	
ผศ.ดร.พิกุลแก้ว	ตังติสานนท์	
ผศ.ดร.สุธีรา	พันธุ์ธีรานุรักษ์	

วัน / เดือน/ ปี ที่สอบ วันศุกร์ที่ 21 กรกฎาคม พ.ศ. 2560 เวลา 09.00-11.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร. คมสัน มาลีสี)
คณบดี คณะวิศวกรรมศาสตร์
วันที่ 21 กรกฎาคม พ.ศ. 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การศึกษาเปรียบเทียบประสิทธิภาพระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบใช้คีย์อ้างอิงสำหรับข้อมูลขนาดใหญ่
นักศึกษา	นายวิทธวัช พวงสายใจ
รหัสประจำตัว	57601304
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2560
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.สุธีรา พันธุ์ธีรานุรักษ์

บทคัดย่อ

ในปัจจุบันปริมาณข้อมูลสารสนเทศมีปริมาณเพิ่มสูงขึ้นอย่างรวดเร็ว และเกิดความหลากหลายเชิงข้อมูลเกิดขึ้น ดังนั้นระบบสารสนเทศจึงจำเป็นจะต้องมีฐานข้อมูลที่มีประสิทธิภาพสูงในการจัดการกับข้อมูลเหล่านี้ ทำให้ระบบฐานข้อมูลแบบไม่สัมพันธ์ได้ถูกคิดค้นขึ้นมาและถูกนำมาพิจารณามากขึ้นในปัจจุบัน แต่อย่างไรก็ตามฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลดังเดิมที่ถูกใช้กันอย่างแพร่หลายในอดีต ดังนั้นการเลือกใช้ฐานข้อมูลให้เหมาะสมจึงเป็นสิ่งสำคัญที่จะช่วยพัฒนาประสิทธิภาพให้กับระบบสารสนเทศได้ ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้ศึกษาเปรียบเทียบประสิทธิภาพการประมวลผลระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เพื่อบ่งชี้การเลือกใช้ฐานข้อมูลให้เหมาะสม โดยฐานข้อมูลเชิงสัมพันธ์ที่นำมาทำการทดลองคือ มาเรียดีบี และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่นำมาทำการทดลองคือ เรดิส ซึ่งทำการทดลองเปรียบเทียบประสิทธิภาพการประมวลผลกับข้อมูล 2 ชุด ได้แก่ ข้อมูลที่มีลักษณะเป็นโครงสร้าง และข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยทำการทดลอง เขียน ลบ แก้ไข และอ่าน จากการเปรียบเทียบผลการทดลองพบว่า ฐานข้อมูลทั้ง 2 ชนิดมีจุดเด่นและจุดด้อยที่แตกต่างกัน โดยฐานข้อมูลเรดิส จะมีจุดเด่นในด้านการเขียน ลบ อ่านข้อมูลที่มีปริมาณมากได้ดีและสามารถรองรับการอ่านข้อมูลที่มีรูปแบบที่ซับซ้อนได้ดี แต่มีจุดอ่อนในด้านการแก้ไขข้อมูล และวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการแปลงข้อมูลที่มีลักษณะโครงสร้างตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์บนฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	A Comparative Study Of Relational Database and Key Value database For Big Data
Student	Mr. Wittawat Puangsaijai
Student ID.	57601304
Degree	Master of Engineering
Program	Information Engineering
Year	2017
Thesis Advisor	Asst.Prof.Dr. Sutheera Puntheeranurak

ABSTRACT

Nowadays, demands of information and more various data are in increasing and growing rapidly. So, it is necessary to have an efficient database system to collect these data. We use a relational database in many applications. Due to more variety data, a relational database may not be the best choice to collect these data. Therefore, NoSQL or non-relational database is introduced for improving the efficiency of an information system. To choose a suitable database for big data application, we compared the performance of key value database management system and relational database management system by using Redis and MariaDB. In the meantime, we proposed the algorithm for transforming relational database into key value database. We experimented with three kinds of operations: insert, update, and delete. The results have shown that Redis has better performance for insertion, delete, and query big data, especially complex query. Moreover, there are weaknesses in an update operation. By the way, this thesis is useful when we have more various data in the system. Our proposed algorithm can be used for database migration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร. สุธีรา พันธุ์ธีรานุรักษ์ ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบคุณเพื่อนๆ ทุกคนที่ได้ช่วยเหลือและเป็นกำลังใจแก่ข้าพเจ้ามาโดยตลอด ทำให้สามารถผ่านพ้นอุปสรรคต่าง ๆ ไปได้ด้วยดี

ท้ายที่สุดนี้ ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และคอยเป็นกำลังใจช่วยเหลือสนับสนุนในทุก ๆด้าน ทำให้ข้าพเจ้าสามารถฟันฝ่าอุปสรรคทุกอย่างไปได้ด้วยดี สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

วิทธวัช พวงสายใจ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 จุดมุ่งหมายและวัตถุประสงค์ของงานวิจัย 4

1.3 สมมติฐานของการวิจัย 4

1.4 ขอบเขตของการวิจัย 4

1.5 ขั้นตอนของการวิจัย 6

1.6 ส่วนประกอบของงานวิจัย 6

บทที่ 2 ทฤษฎีพื้นฐาน 7

2.1 ระบบฐานข้อมูลเชิงสัมพันธ์ 7

2.1.1 ระบบจัดการฐานข้อมูล 7

2.1.2 พื้นฐานการสร้างแบบจำลองข้อมูล 9

2.1.2.1 เอ็นติตี้ 9

2.1.2.2 แอททริบิวต์ 9

2.1.2.3 ความสัมพันธ์ 9

2.1.2.3.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1) 9

2.1.2.3.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (1:M) 9

2.1.2.3.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (M:N) 10

2.1.2.4 ข้อบังคับ 10

2.1.3 แบบจำลองฐานข้อมูล 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.1.4 ฐานข้อมูลเชิงสัมพันธ์ 11
2.1.5 ข้อดีของระบบฐานข้อมูลเชิงสัมพันธ์ 13
2.1.6 ข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์ 13
2.1.7 นิยามคำศัพท์ในระบบฐานข้อมูลเชิงสัมพันธ์ 13
2.1.8 คุณสมบัติ ACID 14
2.1.9 ฐานข้อมูลมาเรียดีบี (MariaDB) 14
2.2 ระบบฐานข้อมูลแบบไม่สัมพันธ์ 15
2.2.1 ทฤษฎี CAP 16
2.2.2 ประเภทของระบบฐานข้อมูลแบบไม่สัมพันธ์ 17
2.2.3 ระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง 20
2.2.4 ฐานข้อมูลเรดิส (Redis) 22
2.2.4.1 คีย์ของฐานข้อมูลเรดิส 23
2.2.4.2 รูปแบบการเก็บข้อมูลแบบ Strings 24
2.2.4.3 รูปแบบการเก็บข้อมูลแบบ List 26
2.2.4.4 รูปแบบการเก็บข้อมูลแบบ Hashes 27
2.2.4.5 รูปแบบการเก็บข้อมูลแบบ Sets 28
2.2.4.6 รูปแบบการเก็บข้อมูลแบบ Sorted sets 30
2.3 งานวิจัยที่เกี่ยวข้อง 31

บทที่ 3 การออกแบบและพัฒนา 36
3.1 การศึกษาและวิเคราะห์ปัญหาของงานวิจัย 36
3.2 การวิเคราะห์และการออกแบบกรอบแนวคิด 38
3.3 ชุดข้อมูลที่ใช้ในงานวิจัย 39
3.3.1 ชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 39
3.3.2 ชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.4 ตัวดำเนินการ 41
3.4.1 การเขียน 41
3.4.1.1 การเขียนสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 42
3.4.1.2 การเขียนสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 42
3.4.2 การลบ 42
3.4.2.1 การลบสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 42
3.4.2.2 การลบสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 43
3.4.3 การแก้ไข 43
3.4.3.1 การแก้ไขสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 43
3.4.3.2 การแก้ไขสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 44
3.4.4 การอ่าน 44
3.4.4.1 การอ่านสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 44
3.4.4.2 การอ่านสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 45
3.5 เครื่องมือในการทดลอง 46
3.5.1 ด้านฮาร์ดแวร์ (Hardware) 46
3.5.2 ด้านซอฟต์แวร์ (Software) 46
3.6 สถิติที่นำมาใช้ในงานวิจัย 46
3.7 การแปลงข้อมูลที่มีลักษณะเป็นตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์ 47
3.7.1 รูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากโครงสร้างตาราง 48
3.7.2 รูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากข้อมูลตาราง 52
3.8 การเปรียบเทียบรูปแบบคำสั่งระหว่างฐานข้อมูลมาเรียดีบีและฐานข้อมูลเรดิส 54

บทที่ 4 การทดลองและผลการทดลอง 56
4.1 ผลการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง 56
4.1.1 ผลการทดลองด้านการเขียนข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.1.2 ผลการทดลองด้านการลบข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง 58

4.1.3 ผลการทดลองด้านการแก้ไขข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง 61

4.1.4 ผลการทดลองด้านการอ่านข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง 63

4.2 ผลการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 67

บทที่ 5 สรุปผลและข้อเสนอแนะ 74

5.1 สรุปผล 74

5.2 ข้อเสนอแนะ 75

เอกสารอ้างอิง 76

ภาคผนวก 80

ภาคผนวก ก. ผลงานทางวิชาการที่ได้รับการตีพิมพ์ในการประชุมทางวิชาการ 81

ประวัติผู้เขียน 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 คำศัพท์ที่ใช้ในระบบฐานข้อมูลเชิงสัมพันธ์ 13

2.2 การเปรียบเทียบตัวอย่างระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละประเภท 19

3.1 สรุปข้อดีและข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์ 36

3.2 สรุปข้อดีและข้อเสียของระบบฐานข้อมูลแบบไม่สัมพันธ์ 37

3.3 ตัวอย่างการเปรียบเทียบคำสั่งสำหรับการอ่าน การลบ การแก้ไข และการอ่านของฐานข้อมูลมาเรียดีบีและฐานข้อมูลเรดิส 54

4.1 ผลการทดลองการเขียนข้อมูลแบบที่ละคำสั่ง 57

4.2 ผลการทดลองการเขียนแบบรวมคำสั่ง 58

4.3 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1 59

4.4 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2 60

4.5 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1 61

4.6 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2 62

4.7 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1 63

4.8 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2 64

4.9 ผลการทดลองการประมวลผลสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 68

4.10 ผลการทดลองการเขียนข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 69

4.11 ผลการทดลองการลบข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 70

4.12 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 71

4.13 ผลการทดลองการอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ระบบจัดการฐานข้อมูล 8

2.2 การโต้ตอบกับระบบจัดการฐานข้อมูลโดยตรง 8

2.3 การโต้ตอบกับระบบจัดการฐานข้อมูลด้วยการผ่านโปรแกรมที่เขียนขึ้น 8

2.4 ตัวอย่างแสดงความสัมพันธ์แบบหนึ่งต่อหนึ่ง 9

2.5 ตัวอย่างแสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม 10

2.6 ตัวอย่างแสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม 10

2.7 ตัวอย่างแสดงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ 12

2.8 คุณสมบัติของระบบฐานข้อมูลแต่ละชนิดที่รองรับทฤษฎี CAP 16

2.9 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง 17

2.10 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสาร 17

2.11 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดกราฟ 18

2.12 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดคอลัมน์ขนาดใหญ่ 18

2.13 รูปแบบการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง 20

2.14 การเข้าถึงข้อมูลแต่ละรูปแบบผ่านค่าของคีย์บนระบบฐานข้อมูลเรดิส 23

2.15 ตัวอย่างการใช้คำสั่ง SET และ GET 24

2.16 ตัวอย่างการใช้คำสั่ง INCR 24

2.17 ตัวอย่างแสดงการใช้คำสั่ง MSET และ MGET 25

2.18 ตัวอย่างแสดงการใช้คำสั่ง EXISTS และ DEL 25

2.19 ตัวอย่างการใช้คำสั่ง EXPIRE 26

2.20 ภาพจำลองการเก็บข้อมูลแบบ List 26

2.21 ตัวอย่างการใช้คำสั่ง LPUSH RPUSH และ LRANGE 27

2.22 ตัวอย่างการใช้คำสั่ง RPOP และ LPOP 27

2.23 ภาพจำลองการเก็บข้อมูลแบบ Hashes 28

2.24 ตัวอย่างการใช้คำสั่ง HMSET HGET และ HMGET 28

2.25 ภาพจำลองการเก็บข้อมูลแบบ Sets 28

2.26 ตัวอย่างการใช้คำสั่ง SADD SMEMBERS และ SISMEMBER 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
2.27	ตัวอย่างการใช้คำสั่ง SUNION SINTER และ SDIFF	29
2.28	ตัวอย่างการใช้คำสั่ง SPOP	30
2.29	ภาพจำลองการเก็บข้อมูลแบบ Sorted sets	30
2.30	ตัวอย่างการใช้คำสั่ง ZADD	31
2.31	ตัวอย่างการใช้คำสั่ง ZRANGE และ ZREVRANGE	31
3.1	กรอบแนวคิดของการวิจัย	38
3.2	แบบจำลองชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง	40
3.3	แบบจำลองชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่	41
3.4	การวัดระยะเวลาในการประมวลผลข้อมูล	47
3.5	การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 1	49
3.6	การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 2 3 และ 4	50
3.7	การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 5 6 7 และ 8	50
3.8	การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 9 10 11 และ 12	51
3.9	ชุดข้อมูลคู่คีย์ผลลัพธ์ที่พิจารณาจากโครงสร้างตาราง	51
3.10	การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 1	52
3.11	การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 2 3 4 และ 5	53
3.12	การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 6	53
4.1	ผลการทดลองการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างแบบทีละคำสั่ง	57
4.2	ผลการทดลองการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างแบบรวมคำสั่ง	58
4.3	ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1	59
4.4	ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2	60
4.5	ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1	61
4.6	ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2	62
4.7	ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1	64
4.8	ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในรูปแบบอย่างง่ายลักษณะที่ 2	65
4.9	ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในรูปแบบซับซ้อนลักษณะที่ 2	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.10 ผลการทดลองการประมวลผลสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 68

4.11 ผลการทดลองการเขียนข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 69

4.12 ผลการทดลองการลบข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 70

4.13 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 71

4.14 ผลการทดลองการอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ระบบสารสนเทศจัดเป็นส่วนประกอบสำคัญในทุกองค์กร โดยประกอบกับระบบฐานข้อมูลซึ่งอยู่ภายใต้โครงสร้างของระบบสารสนเทศซึ่งแทบจะแยกออกจากกันไม่ได้ เนื่องจากระบบฐานข้อมูลเป็นศูนย์รวมข้อมูลในองค์กรและถือเป็นส่วนประกอบสำคัญสำหรับองค์กรที่นำเทคโนโลยีสารสนเทศมาใช้งาน ซึ่งการดำเนินธุรกรรมใด ๆ ของมนุษย์ ก็ล้วนต้องมีส่วนร่วมในการใช้งานระบบฐานข้อมูลไม่ทางตรงก็ทางอ้อม และจากการดำเนินธุรกรรมโดยทั่วไปในยุคนี้ ทุกองค์กรต่างหันมาใช้เทคโนโลยีสารสนเทศเป็นเครื่องมือ ดังนั้นข้อมูลรายการต่าง ๆ ที่เกี่ยวข้องกับการดำเนินธุรกรรมก็จะถูกบันทึกอยู่ในระบบฐานข้อมูล และถูกจัดการอย่างมีระบบ

จากข้อมูลข้างต้น จึงทำให้ทางผู้จัดทำตระหนักได้ว่าระบบฐานข้อมูลจัดเป็นทรัพยากรอันมีค่าอย่างยิ่ง ดังนั้นความท้าทายของผู้ออกแบบระบบฐานข้อมูลก็คือ จะต้องเลือกใช้ระบบฐานข้อมูลชนิดใดเพื่อให้ได้ประสิทธิภาพสูงสุด และจะต้องออกแบบโครงสร้างฐานข้อมูลอย่างไรเพื่อให้ได้มาซึ่งระบบฐานข้อมูลที่เหมาะสมที่สุด สามารถนำมาจัดเก็บข้อมูลต่าง ๆ ภายในองค์กรได้อย่างครบถ้วน ไม่ซ้ำซ้อน และสามารถรองรับการเติบโตได้ในอนาคต ซึ่งสิ่งเหล่านี้จำเป็นต้องพึ่งพานักออกแบบระบบฐานข้อมูลและบุคลากรที่มีความเชี่ยวชาญในเชิง วิชา ทักษะ และประสบการณ์ เพื่อผลักดันให้บรรลุเป้าหมายที่ต้องการได้ในที่สุด

โดยทั่วไปในอดีตระบบฐานข้อมูลที่ถูกใช้กันอย่างแพร่หลาย คือระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) โดยระบบฐานข้อมูลเชิงสัมพันธ์ มีลักษณะโครงสร้างพื้นฐานในรูปแบบของตาราง ที่ประกอบด้วยแถวและสดมภ์ โดยระบบฐานข้อมูลเชิงสัมพันธ์สามารถสั่งการการดำเนิดการต่าง ๆ โดยผ่านภาษาเอสคิวแอล (SQL Query Language) ซึ่งถูกใช้ในการเขียน อ่าน แก้ไขและลบข้อมูล ซึ่งในปัจจุบันเมื่อเทคโนโลยีมีการพัฒนาไปอย่างรวดเร็วและจำนวนข้อมูลข่าวสารในปัจจุบันมีปริมาณที่เพิ่มมากขึ้นในปริมาณมหาศาล รวมไปถึงเกิดความหลากหลายในเชิงของข้อมูลเกิดขึ้น ด้วยเหตุนี้ทำให้ระบบฐานข้อมูลแบบไม่สัมพันธ์ (Non-Relational Database) ถูกพัฒนาขึ้นมาเพื่อตอบสนองความต้องการของผู้ใช้งานระบบฐานข้อมูลในปัจจุบัน

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเป็นยุคข้อมูลข่าวสารและเทคโนโลยีสารสนเทศ ซึ่งมีการพัฒนาเทคโนโลยีต่าง ๆ เป็นไปอย่างรวดเร็ว ส่งผลให้การดำเนิดชีวิตประจำวันของมนุษย์เข้าสู่ยุคแห่งสังคมออนไลน์ จึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งผลให้ข้อมูลข่าวสารบนระบบสารสนเทศในปัจจุบันมีปริมาณเพิ่มขึ้นอย่างรวดเร็ว จนทำให้เกิดข้อมูลจำนวนปริมาณมากหรือเรียกว่า “ข้อมูลขนาดใหญ่” (Big Data) ซึ่งไม่เพียงแต่มีปริมาณข้อมูลที่เพิ่มขึ้นอย่างเดียว แต่ในปัจจุบันข้อมูลข่าวสารยังมีหลากหลายประเภท อาทิเช่น ข้อมูลที่มีโครงสร้าง (Structured Data) ข้อมูลที่ไม่มีโครงสร้าง (Unstructured Data) และข้อมูลที่มีลักษณะกึ่งโครงสร้าง (Semi-Structured Data) โดยข้อมูลข่าวสารในปัจจุบันมีการรับส่งข้อมูลหากันอย่างรวดเร็วหรือเรียกได้ว่ารับส่งข้อมูลกันแบบเรียลไทม์ (Real-Time) ซึ่งข้อมูลเหล่านี้มีทั้งคุณประโยชน์ที่ก่อให้เกิดโอกาสในการดำเนินการทางธุรกิจขององค์กรที่ทำให้เกิดข้อได้เปรียบทางการแข่งขัน ช่วยสนับสนุนการตัดสินใจของผู้บริหารหรือการวางแผนการดำเนินงานทางธุรกิจขององค์กรทั้งในปัจจุบัน และในอนาคต

เนื่องจากข้อมูลข่าวสารในระบบสารสนเทศในปัจจุบันมีปริมาณเพิ่มมากขึ้น ทำให้ส่งผลกระทบต่อประสิทธิภาพต่อระบบสารสนเทศในปัจจุบัน โดยระบบสารสนเทศจะต้องมีการพัฒนาเพื่อเพิ่มขีดความสามารถของระบบสารสนเทศให้สามารถตอบสนองความต้องการต่อผู้ใช้งานระบบได้มากที่สุด ซึ่งหนึ่งในผลกระทบสำคัญนั่นก็คือ ระบบฐานข้อมูล เนื่องจากในอดีตระบบฐานข้อมูลที่ถูกใช้งานกันอย่างแพร่หลายในวงการสารสนเทศคือ ระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งมีการจัดเก็บข้อมูลในรูปแบบของตารางความสัมพันธ์ในลักษณะแถวและสดมภ์ ซึ่งการประมวลผลข้อมูลจำนวนมากอาจทำให้ระบบฐานข้อมูลเชิงสัมพันธ์ประมวลผลช้า และต้องใช้งบประมาณเป็นจำนวนมากในการเพิ่มขีดความสามารถของระบบฐานข้อมูล ดังนั้นระบบฐานข้อมูลแบบไม่สัมพันธ์จึงเข้ามามีบทบาทสำคัญและกำลังเป็นที่น่าสนใจสำหรับองค์กรทางธุรกิจในปัจจุบัน เนื่องจากมีต้นทุนที่ต่ำในการพัฒนา และสามารถจัดการเก็บข้อมูลปริมาณมากได้ดี อย่างเช่น เฟซบุ๊ก (Facebook) กูเกิล (Google) ทวิตเตอร์ (Twitter) เป็นต้น โดยเฉพาะกูเกิลที่มีการพัฒนาระบบฐานข้อมูลแบบไม่สัมพันธ์ขึ้นมาเพื่อให้การค้นหาข้อมูลสามารถกระทำได้อย่างรวดเร็ว ทั้งนี้ระบบฐานข้อมูลแบบไม่สัมพันธ์สามารถแบ่งตามลักษณะการจัดเก็บข้อมูลได้ 4 ประเภทดังนี้ ฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง (Key Value Database) ฐานข้อมูลแบบคอลัมน์ขนาดใหญ่ (Column Oriented Database) ฐานข้อมูลแบบเอกสาร (Document Database)และฐานข้อมูลแบบกราฟ (Graph Database) เนื่องจากระบบฐานข้อมูลแบบไม่สัมพันธ์ไม่ต้องการความสัมพันธ์ใด ๆ ระหว่างข้อมูลทำให้เข้าถึงข้อมูลได้อย่างรวดเร็ว เนื่องจากไม่เน้นเรื่องความเชื่อมั่นของระบบเหมือนระบบฐานข้อมูลเชิงสัมพันธ์ โดยระบบสารสนเทศในปัจจุบันมีการใช้งานระบบฐานข้อมูลที่แตกต่างกัน เนื่องจากระบบฐานข้อมูลแต่ละชนิดมีจุดเด่นและจุดด้อยที่แตกต่างกัน จึงเกิดปัญหาเกิดปัญหาสำหรับนักพัฒนาในการเลือกใช้ระบบฐานข้อมูลที่เหมาะสม และในปัจจุบันระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงเป็นระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลที่ได้รับความนิยมในการประมวลผลข้อมูลขนาดใหญ่และถูกนำไปใช้งานสำหรับจัดการข้อมูลที่มีความหลากหลายได้ดี

ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้ทำการศึกษาออกแบบการทดลอง เพื่อเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์ โดยใช้ระบบจัดการฐานข้อมูลมาเรียดีบี (MariaDB) และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยใช้ระบบจัดการฐานข้อมูลเรดิส (Redis) ซึ่งทางผู้จัดทำได้นำชุดข้อมูล 2 ชุดข้อมูลมาทำการทดลอง โดยชุดข้อมูลชุดแรกคือข้อมูลที่มีลักษณะโครงสร้าง ซึ่งเป็นชุดข้อมูลที่มีความสัมพันธ์กันและชุดข้อมูลชุดที่สองคือ ชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยชุดข้อมูลชุดแรกเป็นชุดข้อมูลที่เหมาะกับระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งทางผู้จัดทำต้องการแสดงให้เห็นถึงประสิทธิภาพของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เมื่อต้องประมวลผลข้อมูลในลักษณะที่มีโครงสร้างและความสัมพันธ์ และเนื่องจากระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เป็นระบบฐานข้อมูลที่มีลักษณะไม่เป็นโครงสร้างและไม่มีความสัมพันธ์ ซึ่งแตกต่างจากระบบฐานข้อมูลเชิงสัมพันธ์โดยสิ้นเชิง ซึ่งระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงยังถือว่าเป็นฐานข้อมูลที่มีรูปแบบการเก็บข้อมูลที่หลากหลายและถือว่าเป็นระบบฐานข้อมูลชนิดหนึ่งที่มีความยืดหยุ่นสูง และมีการเก็บข้อมูลในลักษณะของข้อมูลคู่คีย์ผลลัพธ์ ซึ่งสามารถพัฒนารูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์ให้มีความหลากหลายตามการใช้งานได้ โดยสามารถแบ่งได้ 5 รูปแบบ ประกอบด้วย Sets Lists Sorted Sets Strings และ Hashes เป็นต้น ซึ่งขึ้นอยู่กับผู้ใช้งาน ว่าจะเลือกใช้รูปแบบการเก็บข้อมูลแบบไหนหรือจะออกแบบรูปแบบการเก็บข้อมูลเป็นลักษณะใด ดังนั้นการแปลงข้อมูลที่มีลักษณะที่เป็นโครงสร้างตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์บนระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง จะต้องมีการออกแบบรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์เสียก่อน โดยในงานวิจัยของวิทยานิพนธ์ฉบับนี้ทางผู้จัดทำได้ทำการคิดค้นและออกแบบรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากข้อมูลที่มีลักษณะเป็นโครงสร้างตาราง โดยทางผู้จัดทำได้แบ่งรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์ออกเป็น 2 รูปแบบ ได้แก่ 1. รูปแบบจากโครงสร้างตาราง 2. รูปแบบจากข้อมูลตาราง เพื่อให้สามารถอ้างอิงโครงสร้างความสัมพันธ์และรายละเอียดของข้อมูลไปจัดเก็บบนโครงสร้างของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงได้อย่างถูกต้อง และชุดข้อมูลชุดที่ 2 เป็นชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ ซึ่งข้อมูลชุดนี้เป็นชุดข้อมูลที่มีปริมาณมหาศาลและเป็นข้อมูลที่มีลักษณะของบิ๊กดาต้า โดยทางผู้จัดทำต้องการแสดงให้เห็นถึงประสิทธิภาพของระบบฐานข้อมูลแต่ละชนิด เมื่อนำมาประมวลผลข้อมูลที่มีลักษณะของบิ๊กดาต้า เพราะฉะนั้นจึงออกแบบการทดลองออกเป็น 2 การทดลอง ได้แก่ การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้างและการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยแต่ละการทดลองเปรียบเทียบประสิทธิภาพด้านความเร็วในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลของ 4 ตัวดำเนินการ ได้แก่ ความเร็วในการอ่าน (Select Operation) ความเร็วในการเขียน (Insert Operation) ความเร็วในการแก้ไข (Update Operation) และความเร็วในการลบ (Delete Operation) โดยมีจำนวนข้อมูลและชนิดของข้อมูลเป็นตัวแปรสำคัญในการเปรียบเทียบประสิทธิภาพ เพื่อเป็นประโยชน์แก่ผู้ที่มีส่วนเกี่ยวข้องสามารถนำไปพิจารณาเลือกใช้ระบบฐานข้อมูลที่เหมาะสมในการนำไปประยุกต์ใช้งานกับระบบที่พัฒนาขึ้น

1.2 จุดมุ่งหมายและวัตถุประสงค์ของงานวิจัย

วิทยานิพนธ์ฉบับนี้มุ่งเน้นที่จะศึกษาเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลข้อมูลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เพื่อช่วยเป็นข้อบ่งชี้ถึงความเหมาะสมในการเลือกใช้งานระบบฐานข้อมูล โดยการทดลองในงานวิจัยฉบับนี้แบ่งการทดลองออกเป็น 2 การทดลอง ได้แก่ การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้างและการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยแต่ละการทดลองแบ่งออกเป็น 4 ตัวดำเนินการ คือ การเขียน การลบ การแก้ไขและการอ่านข้อมูล ซึ่งทุกการทดลองจะถูกทดลองบนชุดข้อมูลเดียวกันและทดลองในสภาวะแวดล้อมที่เหมือนกัน และในวิทยานิพนธ์ฉบับนี้ยังได้นำเสนอวิธีการแปลงข้อมูลที่มีลักษณะเป็นโครงสร้างตารางหรือข้อมูลที่มีโครงสร้างความสัมพันธ์ไปเก็บเป็นข้อมูลคู่คีย์ผลลัพธ์บนระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง

1.3 สมมติฐานของการวิจัย

วิทยานิพนธ์ฉบับนี้ได้ตั้งสมมติฐานไว้ว่าการเปรียบเทียบประสิทธิภาพการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง จะสามารถบ่งชี้ถึงจุดเด่นและจุดด้อยของฐานข้อมูลแต่ละชนิดได้ และช่วยให้สามารถพิจารณาเลือกใช้ฐานข้อมูลที่เหมาะสมในการนำไปประยุกต์ใช้งานกับระบบที่พัฒนาขึ้น

1.4 ขอบเขตของการวิจัย

งานวิจัยฉบับนี้เป็นงานวิจัยเชิงทดลอง (Experimental Research) ซึ่งมีขอบเขตการวิจัยดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการออกแบบการทดลองเพื่อเปรียบเทียบประสิทธิภาพความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง ซึ่งแบ่งการทดลองออกเป็น 2 การทดลอง ได้แก่

- การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง
- การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะ ของข้อมูลขนาดใหญ่

ซึ่งแต่ละการทดลองเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลของ 4 ตัว ดำเนินการ

ประกอบด้วย

- การเขียน
- การลบ
- การแก้ไข
- การอ่าน

งานวิจัยฉนับนี้ได้ทำการทดสอบความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยทำการทดลองและประมวลผลแบบเครื่องเดียว โดยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่นำมาใช้ในการทดลองคือ มาเรียดีบี และระบบจัดการฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่นำมาใช้ในการทดลองคือ เรดิส

- ขอบเขตด้านฮาร์ดแวร์ (Hardware) ที่ใช้ในงานวิจัยมีดังนี้
  - หน่วยประมวลผลกลาง (Central Processing Unit : CPU) แบบ Intel Core i5-2450M 2.5 GHz
  - หน่วยความจำหลัก (Random Access Memory : RAM) มีความจุ 8 GB
  - ฮาร์ดดิสก์มีความจุ 1 TB
- ขอบเขตด้านซอฟต์แวร์ (Software) ที่ใช้ในงานวิจัยมีดังนี้
  - ระบบปฏิบัติการ คือ วินโดว์ (Windows) 10 แบบ 64 บิต
  - ฐานข้อมูลเชิงสัมพันธ์ มาเรียดีบี เวอร์ชัน 10.1.13-MariaDB
  - ฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เรดิส เวอร์ชัน 3.2.100.
  - ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรม คือ ภาษาพีเอชพี (PHP)

วิทยานิพนธ์ฉบับนี้ยังได้นำเสนอรูปแบบและวิธีการแปลงข้อมูลที่มีลักษณะเป็นโครงสร้างตารางหรือข้อมูลที่มีโครงสร้างความสัมพันธ์ไปเก็บเป็นข้อมูลคู่คีย์ผลลัพธ์บนระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนของการวิจัย

สามารถแบ่งออกเป็นขั้นตอนดังต่อไปนี้

- ศึกษาถึงแนวโน้มการพัฒนาของระบบฐานข้อมูลในปัจจุบัน
- ศึกษาถึงงานวิจัยที่เกี่ยวข้อง
- ศึกษาถึงคุณลักษณะที่แตกต่างกันระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง
- ศึกษาออกแบบรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากข้อมูลที่มีลักษณะเป็นโครงสร้างตารางหรือข้อมูลที่มีลักษณะโครงสร้างความสัมพันธ์
- ออกแบบโครงสร้างการทดลอง
- ดำเนินการทดลองเพื่อเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง
- ประเมินและวิเคราะห์ผลการทดลอง

1.6 ส่วนประกอบของงานวิจัย

เนื้อหาที่จะกล่าวถึงต่อมาในวิทยานิพนธ์ฉบับนี้ ประกอบด้วย 4 บท คือ

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐาน ซึ่งอธิบายถึงคุณลักษณะของระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์

บทที่ 3 กล่าวถึงการออกแบบและพัฒนาระบบ ซึ่งอธิบายเกี่ยวกับแนวคิดของงานวิจัย ภาพรวมของการออกแบบกระบวนการทดลอง และขั้นตอนการทำงานของการทดลอง รวมถึงข้อมูลที่ใช้ในการทดลอง รวมไปถึงวิธีการแปลงข้อมูลที่มีลักษณะเป็นตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์บนระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง

บทที่ 4 อธิบายการทดลองและผลการทดลอง โดยผลการทดลองด้านความเร็วการประมวลผลที่แสดงถึงประสิทธิภาพของระบบฐานข้อมูลแต่ละชนิด

บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะเกี่ยวข้องกับงานวิจัยที่จะต่อยอดได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐาน

บทนี้จะอธิบายถึงทฤษฎีพื้นฐานที่จำเป็นและสรุปงานวิจัยที่เกี่ยวข้องกับปัญหาที่กำลังศึกษาและวิจัยอยู่ในวิทยานิพนธ์ฉบับนี้ โดยผู้จัดทำวิทยานิพนธ์ฉบับนี้ได้ทำการศึกษาหลักการ ทฤษฎี เทคโนโลยีหรือวิธีการต่าง ๆ ที่จำเป็นต้องใช้สำหรับการออกแบบการทดลอง รวมถึงการทบทวนวรรณกรรมหรืองานวิจัยที่เกี่ยวข้อง เพื่อนำมาประยุกต์ใช้งานกับวิทยานิพนธ์

2.1 ระบบฐานข้อมูลเชิงสัมพันธ์

ในยุคปัจจุบัน คำว่า "ระบบฐานข้อมูล (Database System)" ได้เข้ามามีบทบาทสำคัญต่อวงการระบบเทคโนโลยีสารสนเทศเป็นอย่างมาก ดังจะเห็นได้ว่า แทบทุกที่ที่มีคอมพิวเตอร์ใช้งานเพื่อประมวลผล ไม่ว่าจะเป็นการประยุกต์ใช้คอมพิวเตอร์สำหรับงานด้านธุรกิจ การศึกษา การแพทย์ วิทยาศาสตร์ หรืองานด้านวิศวกรรม ก็ล้วนต้องเกี่ยวข้องกับระบบฐานข้อมูลทั้งสิ้น ดังนั้นจึงอาจจะกล่าวได้ว่าอัตราการเติบโตของการใช้งานคอมพิวเตอร์ ได้ส่งผลกระทบโดยตรงต่อเทคโนโลยีระบบฐานข้อมูล จึงส่งผลให้แต่ละองค์กรต่างจำเป็นต้องใช้งานคอมพิวเตอร์เพื่อการเข้าถึงระบบฐานข้อมูลที่ตนต้องการ

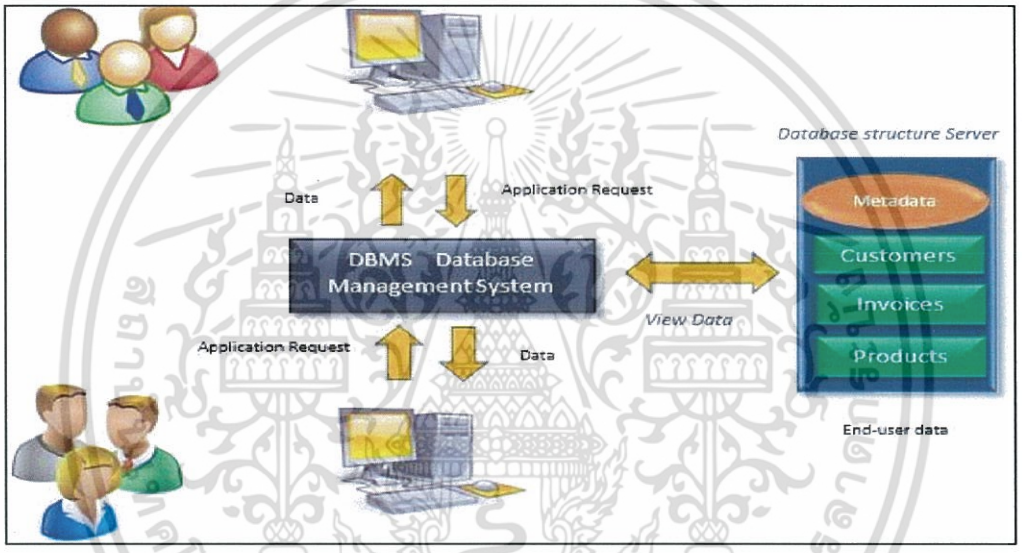
ระบบฐานข้อมูลคือศูนย์รวมของข้อมูลต่าง ๆ ที่มีความสัมพันธ์กัน (Relationship) โดยจะมีกระบวนการในการจัดหมวดหมู่ข้อมูลอย่างมีระเบียบแบบแผน ก่อให้เกิดระบบฐานข้อมูลที่เป็นแหล่งรวมของข้อมูลจากแหล่งต่าง ๆ ซึ่งจะถูกเก็บไว้อย่างมีระบบภายในระบบฐานข้อมูล โดยผู้ใช้งานสามารถเข้าถึงข้อมูลส่วนกลางนี้เพื่อนำไปประมวลผลร่วมกันได้ จากตรงจุดนี้ทำให้ตระหนักถึงความสำคัญของระบบฐานข้อมูลได้ว่า ระบบฐานข้อมูลจะเป็นหัวใจสำคัญสำหรับองค์กรทุกองค์กรที่ต้องการเติบโตทางธุรกิจและเป็นหัวใจสำคัญในการแข่งขันทางธุรกิจในอนาคต [31]

2.1.1 ระบบจัดการฐานข้อมูล

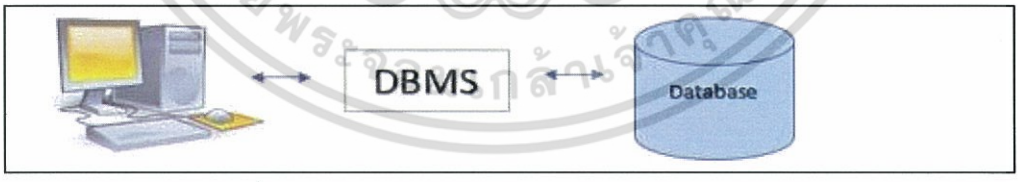
ระบบจัดการฐานข้อมูล (Database Management Systems, ระบบจัดการฐานข้อมูล) [31] คือซอฟต์แวร์ที่ใช้เป็นเครื่องมือของผู้ใช้เพื่อโต้ตอบกับระบบฐานข้อมูล ซึ่งระบบจัดการฐานข้อมูล จะประกอบไปด้วยฟังก์ชันหน้าที่ต่าง ๆ ในการจัดการกับข้อมูล รวมทั้งภาษาที่ใช้ทำงานกับข้อมูล ซึ่งโดยทั่วไปมักใช้ภาษาเอสคิวแอล (SQL) ในการโต้ตอบระหว่างกันกับผู้ใช้ด้วยการสร้าง การเรียกดู และการบำรุงรักษาระบบฐานข้อมูล นอกจากนี้ระบบจัดการฐานข้อมูลยังมีหน้าที่ในการรักษาความมั่นคงและปลอดภัยของข้อมูล ด้วยการป้องกันมิให้ผู้ไม่มีสิทธิ์การใช้งานเข้ามาละเมิดข้อมูลในระบบฐานข้อมูลที่เป็นศูนย์กลางได้ รวมถึงการสำรองข้อมูลและการกู้คืนข้อมูล ในกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

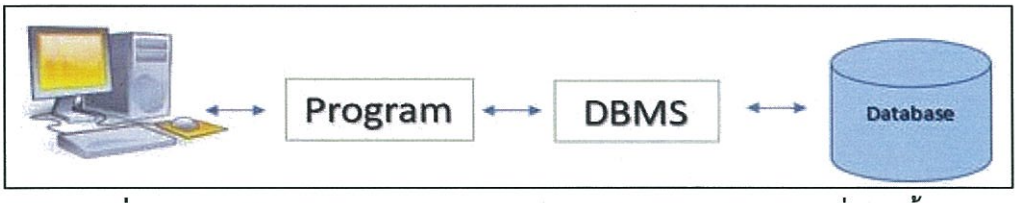
ข้อมูลเกิดความเสียหาย เป็นต้น ซึ่งระบบจัดการฐานข้อมูล ได้แสดงดังรูปที่ 2.1 จึงกล่าวโดยสรุปว่า ระบบจัดการฐานข้อมูล เป็นซอฟต์แวร์หรือโปรแกรมที่ใช้สำหรับโต้ตอบกับผู้ใช้งาน โดย ระบบจัดการฐานข้อมูล จะเป็นตัวกลางในการโต้ตอบระหว่างผู้ใช้งานกับระบบฐานข้อมูล ซึ่งผู้ใช้สามารถโต้ตอบกับระบบฐานข้อมูล ระบบจัดการฐานข้อมูล โดยตรง ดังแสดงในรูปที่ 2.2 หรือผ่านโปรแกรมประยุกต์ก็ได้ เช่น การสร้างโปรแกรมด้วยภาษาจาวา (Java) เพื่อโต้ตอบกับ ระบบจัดการฐานข้อมูล ที่ใช้งานดังแสดงในรูปที่ 2.3 โดยทั่วไป ระบบจัดการฐานข้อมูล ที่นิยมใช้กันในปัจจุบัน ยกตัวอย่างเช่น มายเอสคิวแอล (MySQL) ออราเคิล (Oracle) มาเรียดีบี (MariaDB) เอสคิวแอล เซิร์ฟเวอร์ (SQL Server) และโพสต์เกรสคิวเอล (PostgreSQL) เป็นต้น



รูปที่ 2.1 ระบบจัดการฐานข้อมูล



รูปที่ 2.2 การโต้ตอบกับระบบจัดการฐานข้อมูลโดยตรง



รูปที่ 2.3 การโต้ตอบกับระบบจัดการฐานข้อมูลด้วยการผ่านโปรแกรมที่เขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 พื้นฐานการสร้างแบบจำลองข้อมูล

พื้นฐานการสร้างแบบจำลองข้อมูล (Data Model Basic Building Blocks) โดยทั่วไปจะประกอบไปด้วยเอ็นติตี้ แอททริบิวต์ ความสัมพันธ์ และข้อบังคับ [31]

2.1.2.1 เอ็นติตี้

เอ็นติตี้ (Entities) คือ ทุก ๆ สิ่ง ไม่ว่าจะเป็นบุคคล สถานที่ สิ่งของ หรือเหตุการณ์ที่เกี่ยวข้องกับข้อมูลที่รวบรวมไว้เพื่อการจัดเก็บ เอ็นติตี้ถือเป็นตัวแทนของวัตถุในโลกแห่งความเป็นจริง ตัวอย่างเช่น เอ็นติตี้พนักงาน เอ็นติตี้รถยนต์ นอกจากนี้แล้วเอ็นติตี้ยังอาจเป็นนามธรรม (Abstractions) ก็ได้ เช่น เอ็นติตี้เที่ยวบิน เป็นต้น

2.1.2.2 แอททริบิวต์

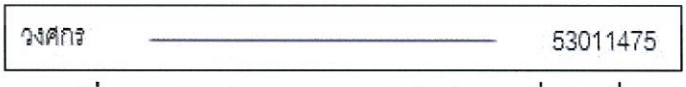
แอททริบิวต์ (Attributes) คือ คุณลักษณะของเอ็นติตี้ ตัวอย่างเช่น เอ็นติตี้พนักงาน ที่ประกอบไปด้วยแอททริบิวต์รหัสพนักงาน ชื่อ-นามสกุล ที่อยู่ โทรศัพท์ เงินเดือน เป็นต้น

2.1.2.3 ความสัมพันธ์

ความสัมพันธ์ (Relationships) คือ ความสัมพันธ์ระหว่างเอ็นติตี้ ตัวอย่างเช่น ความสัมพันธ์ระหว่างลูกค้ากับบัญชีธนาคาร เช่น ลูกค้าสามารถเปิดบัญชีธนาคารได้หลายบัญชี เป็นต้น ซึ่งแบบจำลองข้อมูลจะมีความสัมพันธ์อยู่ 3 รูปแบบด้วยกัน ได้แก่ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง ความสัมพันธ์แบบหนึ่งต่อกลุ่มและความสัมพันธ์แบบกลุ่มต่อกลุ่ม ซึ่งโดยปกติแล้วนักออกแบบระบบฐานข้อมูลมักใช้สัญลักษณ์ย่อเพื่อความสะดวกต่อการใช้งานคือ 1:1 1:M และ M:N ตามลำดับ และต่อไปนี้เป็นรายละเอียดของความแตกต่างด้านความสัมพันธ์ทั้งสามชนิด

2.1.2.3.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1)

ความสัมพันธ์แบบหนึ่งต่อหนึ่งเป็นความสัมพันธ์ที่แต่ละรายการของเอ็นติตี้ที่หนึ่งมีความสัมพันธ์กับเอ็นติตี้ที่สองเพียงหนึ่งรายการ ตัวอย่างเช่น นักศึกษามีรหัสประจำตัวได้เพียงหนึ่งรหัส และรหัสนักศึกษาหนึ่งรหัสจะนำไปอ้างอิงนักศึกษาได้เพียงหนึ่งคนเท่านั้นดังแสดงในรูปที่ 2.4



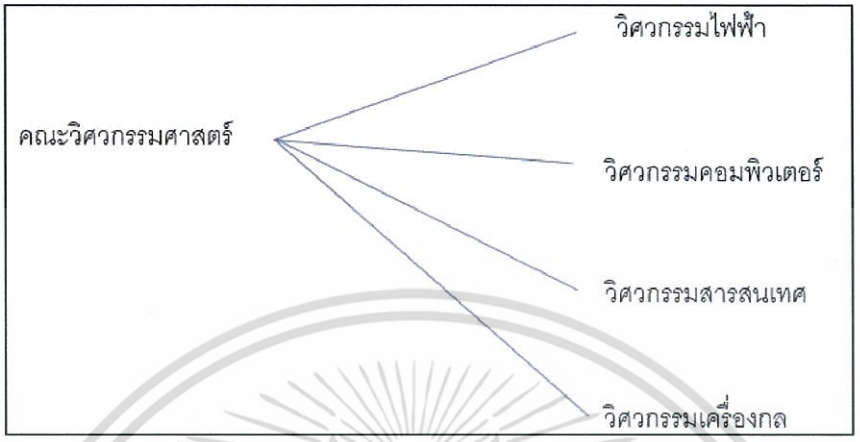
รูปที่ 2.4 ตัวอย่างแสดงความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2.1.2.3.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (1:M)

ความสัมพันธ์แบบหนึ่งต่อกลุ่มเป็นความสัมพันธ์ที่แต่ละรายการของเอ็นติตี้ที่หนึ่ง มีความสัมพันธ์กับเอ็นติตี้ที่สองมากกว่าหนึ่งรายการ โดยแต่ละรายการของเอ็นติตี้ที่สอง จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

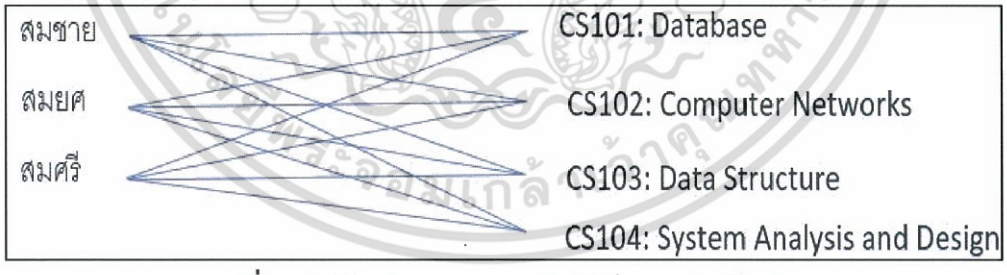
ความสัมพันธ์กับเอ็นติตี้ที่หนึ่ง ได้เพียงหนึ่งรายการเท่านั้น ตัวอย่างเช่น คณะหนึ่งมีหลายสาขาวิชา ในขณะที่แต่ละสาขาวิชาจะต้องสังกัดเพียงหนึ่งคณะดังแสดงในรูปที่ 2.5



รูปที่ 2.5 ตัวอย่างแสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม

2.1.2.3.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (M:N)

ความสัมพันธ์แบบกลุ่มต่อกลุ่มเป็นความสัมพันธ์ที่แต่ละรายการของเอ็นติตี้ที่หนึ่งมีความสัมพันธ์กับข้อมูลเอ็นติตี้ที่สองมากกว่าหนึ่งรายการ ในขณะที่แต่ละรายการของเอ็นติตี้ที่สองก็มีความสัมพันธ์กับข้อมูลที่หนึ่ง ได้มากกว่าหนึ่งรายการเช่นกัน ตัวอย่างเช่น นักศึกษาลงทะเบียนได้หลายวิชา โดยแต่ละวิชาก็มีนักศึกษาลงทะเบียนได้มากกว่าหนึ่งคนดังแสดงในรูปที่ 2.6



รูปที่ 2.6 ตัวอย่างแสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม

2.1.2.4 ข้อบังคับ

ข้อบังคับ คือ กฎเกณฑ์เพื่อการบรรจุข้อมูล ซึ่งกฎข้อบังคับนี้ถือว่าเป็นสิ่งสำคัญ เพราะจะช่วยให้เกิดความเป็นอันหนึ่งอันเดียวกัน เพื่อให้เกิดความสอดคล้องตรงกันของข้อมูล ตัวอย่างเช่น เงินเดือนพนักงานจะต้องมีค่าระหว่าง 9,000 ถึง 300,000 บาท หรือนักศึกษาสามารถลงทะเบียนเรียนได้หลายหน่วยกิต แต่รวมแล้วต้องไม่เกิน 18 หน่วยกิต เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 แบบจำลองฐานข้อมูล

ระบบสารสนเทศยุคใหม่ได้มีการนำเทคโนโลยีระบบฐานข้อมูล เพื่อนำไปใช้จัดการระบบฐานข้อมูลให้เกิดประโยชน์ในด้านการจัดเก็บข้อมูลและการเรียกดูข้อมูล ซึ่งความหมายของระบบฐานข้อมูล ก็คือศูนย์รวมของไฟล์ข้อมูลที่มีความสัมพันธ์กันอยู่ โดยความสัมพันธ์ระหว่างไฟล์ภายในระบบฐานข้อมูลนั้นจะเชื่อมโยงความสัมพันธ์อย่างมีระบบ ซึ่งเป็นไปตามแนวความคิดของแบบจำลองข้อมูลเชิงตรรกะ ความสัมพันธ์ในข้อมูลดังกล่าวทำให้สามารถเข้าถึงข้อมูลในระบบฐานข้อมูลที่แตกต่างกัน ดังนั้นกระบวนการคัดสรรระบบจัดการฐานข้อมูล เพื่อใช้งานในขั้นตอนการออกแบบระบบฐานข้อมูล จึงจำเป็นต้องสอดคล้องกับแบบจำลองข้อมูลที่สร้างขึ้นจากระยะการวิเคราะห์ด้วย

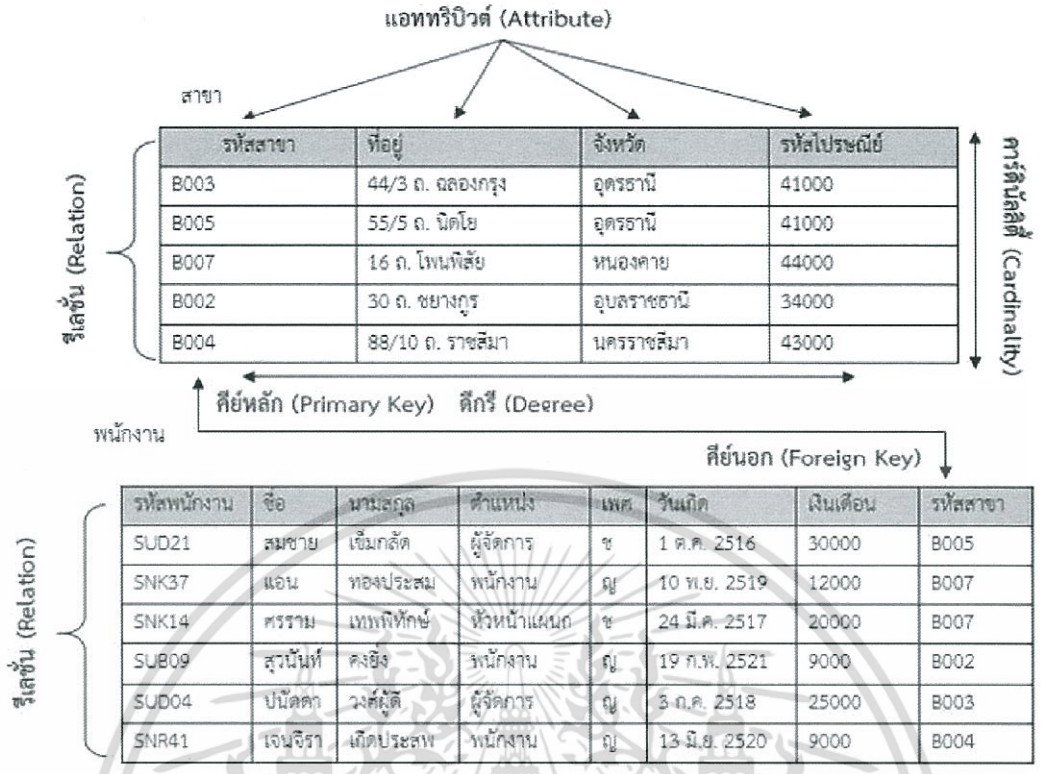
โครงสร้างรูปแบบการจัดการระบบฐานข้อมูลมีหลายรูปแบบด้วยกัน ซึ่งแต่ละรูปแบบต่างก็มีคุณสมบัติและโครงสร้างที่แตกต่างกัน โดยรายละเอียดการจัดการระบบฐานข้อมูลจะต้องสนับสนุนหรือตั้งอยู่บนพื้นฐานของแบบจำลองระบบฐานข้อมูล 1 ใน 5 แบบ ซึ่งจะกล่าวในรายละเอียดดังนี้ [31]

- แบบจำลองระบบฐานข้อมูลลำดับชั้น
- แบบจำลองระบบฐานข้อมูลเครือข่าย
- แบบจำลองระบบฐานข้อมูลเชิงสัมพันธ์
- แบบจำลองระบบฐานข้อมูลเชิงวัตถุ
- แบบจำลองระบบฐานข้อมูลแบบมัลติไดเมนชัน

2.1.4 ฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) [31] เป็นฐานข้อมูลที่ใช้แบบจำลองเชิงสัมพันธ์ (Relational Database Model) ซึ่งเป็นแบบจำลองที่ได้รับความนิยมสูงสุดและใช้กันอย่างแพร่หลายในปัจจุบัน สาเหตุที่เป็นเช่นนั้นเนื่องจากแบบจำลองฐานข้อมูลเชิงสัมพันธ์นี้นำเสนอมุมมองของข้อมูลในลักษณะตาราง ซึ่งสามารถสื่อสัมพันธ์กับมนุษย์ได้เข้าใจง่าย โดยตารางหนึ่งจะประกอบด้วยแถวและสดมภ์ ข้อมูลที่จัดเก็บอยู่ในตารางสามารถเชื่อมโยงความสัมพันธ์กับตารางอื่น ๆ ได้ ไม่ว่าจะเป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มหรือแบบกลุ่มต่อกลุ่ม โดยจะใช้คีย์ในการอ้างอิงถึงตารางอื่น ๆ ที่เกี่ยวข้อง ซึ่งคีย์ดังกล่าวยังสามารถเป็นได้ทั้งคีย์หลัก (Primary Key) และคีย์รอง (Secondary Key) รวมถึงการกำหนดลำดับดัชนีเพื่อเข้าถึงข้อมูลได้อย่างรวดเร็ว ซึ่งมีส่วนประกอบโครงสร้างพื้นฐานของฐานข้อมูลเชิงสัมพันธ์ประกอบด้วยรีเลชัน (Relation) แอททริบิวต์ (Attribute) ทูเพิล (Tuple) โดเมน (Domain) ดีกรี (Degree) คาร์ดินัลลิตี้ (Cardinality) คีย์หลัก (Primary Key)และคีย์นอก (Foreign Key) ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รหัสสาขา	ที่อยู่	จังหวัด	รหัสไปรษณีย์
B003	44/3 ถ. ฉลองกรุง	อุดรธานี	41000
B005	55/5 ถ. นิดโย	อุดรธานี	41000
B007	16 ถ. โพนพิสัย	หนองคาย	44000
B002	30 ถ. ชยางกูร	อุบลราชธานี	34000
B004	88/10 ถ. ราชสีมา	นครราชสีมา	43000

รหัสพนักงาน	ชื่อ	นามสกุล	ตำแหน่ง	เพศ	วันเกิด	เงินเดือน	รหัสสาขา
SUD21	สมชาย	เข็มกลัด	ผู้จัดการ	ช	1 ต.ค. 2516	30000	B005
SNK37	แอน	ทองประสม	พนักงาน	ญ	10 พ.ย. 2519	12000	B007
SNK14	ศรราม	เทพพิทักษ์	หัวหน้าแผนก	ช	24 มี.ค. 2517	20000	B007
SUB09	สุวนันท์	คงยิ่ง	พนักงาน	ญ	19 ก.พ. 2521	9000	B002
SUD04	ปนัดดา	วงศ์ผู้ดี	ผู้จัดการ	ญ	3 ก.ค. 2518	25000	B003
SNR41	เจนจิรา	เกิดประสพ	พนักงาน	ญ	13 มิ.ย. 2520	9000	B004

รูปที่ 2.7 ตัวอย่างแสดงโครงสร้างฐานข้อมูลเชิงสัมพันธ์

จากรูปที่ 2.7 แสดงส่วนประกอบโครงสร้างพื้นฐานของฐานข้อมูลเชิงสัมพันธ์ซึ่งประกอบด้วย 8 ส่วนประกอบสำคัญดังนี้ [31]

1. รีเลชัน หมายถึงตารางสองมิติ ซึ่งประกอบด้วยสดมภ์และแถว ซึ่งมีความสัมพันธ์กันของข้อมูล

2. แอททริบิวต์ หมายถึงชื่อสดมภ์ต่าง ๆ ในรีเลชันที่ใช้อธิบายหรือกำหนดคุณลักษณะของแต่ละตารางหรือเอ็นติตี้

3. ทูเพิลหรือแถว หมายถึงแถวของรีเลชัน

4. โดเมน หมายถึงเป็นการกำหนดขอบเขตค่าข้อมูล และรูปแบบชนิดข้อมูลของแต่ละแอททริบิวต์

5. ดีกรี หมายถึงจำนวนแอททริบิวต์ที่บรรจุอยู่ในรีเลชัน

6. คาร์ดินัลลิตี้ หมายถึงจำนวนของทูเพิลหรือจำนวนแถวในตารางที่บรรจุอยู่

7. คีย์หลัก หมายถึงแอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่ถูกคัดเลือกให้เป็นคีย์หลัก ที่สามารถนำไปใช้เพื่อการอ้างอิงความเป็นเอกลักษณ์ของแต่ละแถวในตารางนั้น ๆ ได้ โดยคีย์หลักต้องไม่บรรจุข้อมูลว่างเปล่า (Null)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.) คีย์นอก หมายถึงแอทททริบิวต์หรือกลุ่มของแอททริบิวต์ในตารางหนึ่งที่มีคุณสมบัติเป็นคีย์หลัก และไปปรากฏในอีกตารางหนึ่งเพื่อใช้สำหรับการเชื่อมโยงระหว่างกัน ซึ่งเป็นคีย์ที่ใช้สำหรับเชื่อมโยงความสัมพันธ์ระหว่างตาราง

2.1.5 ข้อดีของระบบฐานข้อมูลเชิงสัมพันธ์

1. มีความเป็นอิสระในโครงสร้าง โดยเมื่อมีการเปลี่ยนแปลงโครงสร้างของตาราง จะไม่ส่งผลต่อแอปพลิเคชันโปรแกรมที่ใช้งาน

2. การนำเสนอข้อมูลในรูปแบบของตาราง ทำให้สามารถมองเห็นภาพการเก็บข้อมูลได้ชัดเจนมากขึ้น ทำให้ง่ายต่อการออกแบบฐานข้อมูล การนำไปใช้ และการจัดการ

3. การเรียกดูข้อมูล สามารถเรียกใช้ได้ด้วยชุดคำสั่งเอสคิวแอล

4. มีระบบความปลอดภัยที่ดี เนื่องจากโครงสร้างนี้ผู้ใช้งานจะไม่ทราบถึงกระบวนการจัดเก็บข้อมูลภายในฐานข้อมูลแท้จริงว่าเป็นอย่างไร

5. ระบบจัดการฐานข้อมูลที่พัฒนาขึ้นมาในปัจจุบันล้วนรองรับเทคโนโลยีฐานข้อมูลเชิงสัมพันธ์

2.1.6 ข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์

1.) ระบบฐานข้อมูลจะมีประสิทธิภาพลดลงเมื่อ แบบจำลองของระบบฐานข้อมูลมีความซับซ้อน

2.) จำเป็นต้องเสียค่าใช้จ่ายในระบบค่อนข้างสูง เนื่องจากทรัพยากรทั้งตัวฮาร์ดแวร์และซอฟต์แวร์ที่นำมาใช้ต้องมีความสามารถสูง

3.) เกิดการสูญเสียข้อมูลได้ เนื่องจากข้อมูลต่าง ๆ ภายในฐานข้อมูลจะถูกจัดเก็บอยู่ในที่เดียวกัน ดังนั้นถ้าที่เก็บข้อมูลเกิดมีปัญหา อาจทำให้ต้องสูญเสียข้อมูลทั้งหมดในฐานข้อมูลได้ ดังนั้นการจัดทำฐานข้อมูลที่ดีจึงต้องมีการสำรองข้อมูลไว้เสมอ

2.1.7 นิยามคำศัพท์ในระบบฐานข้อมูลเชิงสัมพันธ์

นิยามคำศัพท์ที่เรียกใช้แทนกันในระบบฐานข้อมูลเชิงสัมพันธ์ซึ่งได้มีการเปรียบเทียบกับคำศัพท์ทั่วไปที่พบเจอบ่อย ๆ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 คำศัพท์ที่ใช้ในระบบฐานข้อมูลเชิงสัมพันธ์

คำศัพท์ทางวิชาการ	คำศัพท์ทั่วไป
รีเลชัน	ตาราง
ทูเพิล	แถว หรือ เรคคอร์ด
แอททริบิวต์	สดมภ์ หรือ ฟิลด์
คาร์ดินาลลิตี้	จำนวนแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 คำศัพท์ที่ใช้ในระบบฐานข้อมูลเชิงสัมพันธ์ (ต่อ)

คำศัพท์ทางวิชาการ	คำศัพท์ทั่วไป
คีย์หลัก	ค่าเอกลักษณ์
โดเมน	ขอบเขตค่าของข้อมูล

2.1.8 คุณสมบัติ ACID

เป็นคุณสมบัติของระบบฐานข้อมูลเชิงสัมพันธ์ เพื่อสร้างความน่าเชื่อถือให้กับระบบงานมากยิ่งขึ้น และแต่ละคำมีความหมายดังต่อไปนี้ [32]

1. Atomicity หมายถึง รายการ (Transaction) จะไม่สามารถแบ่งแยกงานออกเป็นส่วนย่อย ๆ ได้ ในการทำการประมวลผลรายการ โดยถ้าสำเร็จจะต้องสำเร็จทั้งหมด แต่ถ้าไม่สำเร็จจะต้องไม่สำเร็จทั้งหมด

2. Consistency หมายถึง ฐานข้อมูลจะต้องอยู่ในสภาพที่สอดคล้องและถูกต้องเสมอ การประมวลผลรายการจะต้องไม่ทำให้ข้อมูลในฐานข้อมูลเกิดความผิดพลาด โดยหลังจากการประมวลผลรายการเสร็จสิ้น กระบวนการทำงานก็จะออกจากระบบ โดยอาจจะเป็นสถานะที่ถูกต้องหรือสถานะล้มเหลว ซึ่งถ้าหากการประมวลผลรายการล้มเหลวหรือไม่สามารถบรรลุถึงสถานะจุดสิ้นสุดที่มีเสถียรภาพหรือสถานะที่ถูกต้องได้ ระบบจะต้องนำระบบกลับไปสู่สถานะเริ่มต้นใหม่

3. Isolation หมายถึง พฤติกรรมต่าง ๆ ของการประมวลผลรายการ จะไม่ส่งผลกระทบต่อการประมวลผลรายการอื่นที่ทำงานพร้อมกันในขณะนั้น และจะต้องมีลำดับการเข้าใช้งานทรัพยากรร่วมกัน ซึ่งมีการรับรองว่าต้องไม่ส่งผลกระทบต่อกันและกัน โดยโปรแกรมแบบหลายผู้ใช้งานที่ประมวลผลภายใต้ข้อมูลที่มีการป้องกัน ต้องกระทำตัวเหมือนอยู่ในสภาพแวดล้อมที่มีผู้ใช้งานแบบคนเดียว ผลลัพธ์จากการที่ใช้ทรัพยากรร่วมกันนั้นจะไม่แสดงในระหว่างการทำงานหรือก่อนการยืนยัน

4. Durability หมายถึง การประมวลผลรายการใด เมื่อสั่งการประมวลผลไปแล้ว ระบบจะต้องทำงานให้เสร็จสิ้น และข้อมูลจะต้องไม่เกิดการสูญหาย ถึงแม้ว่าระบบจะมีเหตุขัดข้อง เช่น เครื่องหยุดการทำงานเนื่องจากไฟฟ้าดับ ระบบจัดการระบบฐานข้อมูลจะต้องรักษาสถานะของระบบฐานข้อมูลที่ถูกต้องไว้เสมอ

2.1.9 ฐานข้อมูลมาเรียดีบี (MariaDB)

ฐานข้อมูลมาเรียดีบี เปรียบเสมือนเป็นฐานข้อมูลรุ่นหนึ่งของฐานข้อมูลมายเอสคิวแอล ซึ่งเป็นที่ทราบกันดีว่าฐานข้อมูลมายเอสคิวแอล เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่ได้รับความนิยมมากที่สุดในโลก เนื่องจากในปี 2008 บริษัท ซัน ไมโครซิสเต็มส์ (Sun Microsystems) ได้ทำการเข้าซื้อฐานข้อมูลมายเอสคิวแอล หลังจากนั้นกระบวนการพัฒนาหรือทีมนักพัฒนาฐานข้อมูลมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอสคิวแอล จึงได้เปลี่ยนแปลงไป โดยฐานข้อมูลมายเอสคิวแอล ได้ถูกนำไปพัฒนาน้อยลง ดังนั้นในปี 2009 นาย Michael Widenius ซึ่งเป็นผู้ก่อตั้งฐานข้อมูลมายเอสคิวแอลเดิม จึงได้แยกตัวหรือแยกโครงการออกมาเพื่อพัฒนาฐานข้อมูลชนิดใหม่ นั่นก็คือฐานข้อมูลมาเรียดีบีนั่นเอง [28] โดยโครงสร้างต่าง ๆ ของฐานข้อมูลมาเรียดีบี สามารถเรียกได้ว่าเหมือนกับโครงสร้างฐานข้อมูลมายเอสคิวแอลทั้งหมด ดังนั้นจึงสามารถแทนที่ฐานข้อมูลมายเอสคิวแอล ด้วยฐานข้อมูลมาเรียดีบีได้เลยแบบไม่ต้องปรับเปลี่ยนอะไรทั้งสิ้น ซึ่งประสิทธิภาพของฐานข้อมูลมาเรียดีบี สามารถทำงานได้เร็วกว่าฐานข้อมูลมายเอสคิวแอล อยู่ 10 เปอร์เซ็นต์ โดยในปัจจุบันบริษัทหรือเว็บไซต์ต่าง ๆ ได้ทยอยเปลี่ยนจากฐานข้อมูลมายเอสคิวแอลเป็นฐานข้อมูลมาเรียดีบีเรียบร้อยแล้ว อาทิเช่น วิกิพีเดีย (Wikipedia) ได้เลือกใช้ฐานข้อมูลมาเรียดีบีเป็นฐานข้อมูลหลักแทนการใช้ฐานข้อมูลมายเอสคิวแอล และบริษัทเรดแฮต (Red Hat) ได้ประกาศว่า Red Hat Enterprise(RHEL) เวอร์ชันถัดไปจะเปลี่ยนไปใช้ฐานข้อมูลมาเรียดีบี เป็นฐานข้อมูลหลักแทนการใช้ฐานข้อมูลเดิมซึ่งก็คือมายเอสคิวแอล[29]

โดยฐานข้อมูลมาเรียดีบี สามารถเก็บข้อมูลในลักษณะที่เป็นโครงสร้างและสามารถเก็บข้อมูลในลักษณะอาร์เรย์ขนาดใหญ่ได้ รวมถึงเหมาะสำหรับนำไปเก็บข้อมูลที่เกี่ยวข้องกับระบบธนาคาร เพราะฐานข้อมูลมาเรียดีบี นั้นได้ถูกพัฒนาอยู่เสมอและเป็นฐานข้อมูลที่มีประสิทธิภาพดีกว่าฐานข้อมูลมายเอสคิวแอล ซึ่งสามารถกล่าวได้ว่าฐานข้อมูลมาเรียดีบี จะมาแทนที่ฐานข้อมูลมายเอสคิวแอลนั่นเอง โดยประสิทธิภาพที่นักพัฒนาได้ทำการทดสอบ พบว่าฐานข้อมูลมาเรียดีบีเป็นฐานข้อมูลที่มีความเร็วในการตอบสนอง สามารถรองรับการขยายขนาดระบบฐานข้อมูลได้ดีและเป็นฐานข้อมูลที่มีความทนทานสูง และข้อดีอีกหนึ่งข้อคือ มีเครื่องมือหรือปลั๊กอินจำนวนมากที่สามารถนำมาใช้งานร่วมกับฐานข้อมูลมาเรียดีบี ทำให้ฐานข้อมูลนี้มีความหลากหลายในการพัฒนามากยิ่งขึ้น ฐานข้อมูลมาเรียดีบี ถูกพัฒนาและถูกนำไปใช้งานในวงการโอเพนซอร์ซอย่างแพร่หลาย และฐานข้อมูลมาเรียดีบี มีลักษณะโครงสร้างเป็นระบบฐานข้อมูลเชิงสัมพันธ์ โดยใช้ภาษาเอสคิวแอลในการเข้าถึงข้อมูลหรือเรียกดูข้อมูลในระบบฐานข้อมูล และในเวอร์ชันรุ่นล่าสุดของฐานข้อมูลมาเรียดีบี นั้นสามารถรองรับการเก็บข้อมูลได้หลากหลายรูปแบบมากขึ้น ยกตัวอย่าง สามารถเก็บข้อมูลที่มีลักษณะเป็นระบบสารสนเทศภูมิศาสตร์ (GIS Data) และข้อมูลที่มีลักษณะของเจซันฟอร์แมต (JSON format data)ได้

2.2 ระบบฐานข้อมูลแบบไม่สัมพันธ์

ฐานข้อมูลแบบไม่สัมพันธ์ (Non-Relational Database) เป็นรูปแบบระบบฐานข้อมูลที่มีแนวคิดตรงข้ามกับระบบฐานข้อมูลเชิงสัมพันธ์โดยสิ้นเชิง เนื่องจากระบบฐานข้อมูลแบบไม่สัมพันธ์จะไม่มีความสัมพันธ์ระหว่างข้อมูล และใช้ภาษาในการทำงานแบบไม่มีโครงสร้างในการจัดการกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล เน้นการเข้าถึงที่รวดเร็ว และโครงสร้างถูกออกแบบมาเพื่อรองรับการทำงานร่วมกับข้อมูลที่มีปริมาณมากและรองรับการขยายตัวแบบกระจายได้ โดยระบบฐานข้อมูลแบบไม่สัมพันธ์บางประเภทหรือบางยี่ห้อไม่รับรองความถูกต้องของข้อมูลเหมือนกับคุณสมบัติ ACID ของระบบฐานข้อมูลเชิงสัมพันธ์ แต่ระบบฐานข้อมูลแบบไม่สัมพันธ์รับรองคุณสมบัติของทฤษฎี CAP โดยทฤษฎี CAP [23] ประกอบด้วยคุณสมบัติ Consistency Availability และ Partition Tolerance ซึ่งระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละชนิดสามารถรับรองคุณสมบัติของทฤษฎี CAP ได้เพียง 2 ใน 3 คุณสมบัติเท่านั้น ขึ้นอยู่กับจุดเด่นและจุดด้อยของระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละชนิด

2.2.1 ทฤษฎี CAP

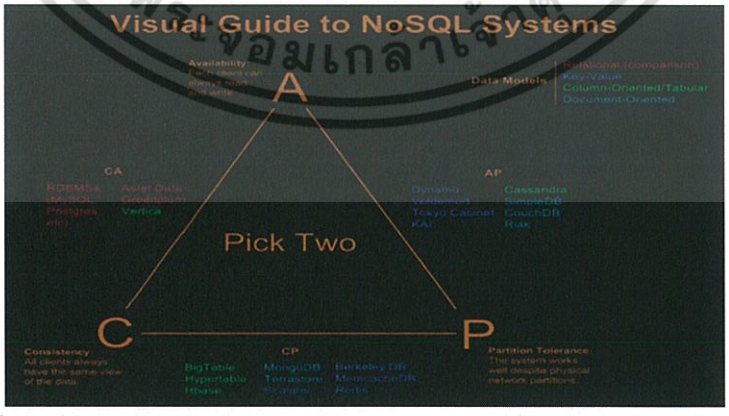
โดยทฤษฎี CAP [13 , 23] ประกอบด้วย 3 คุณสมบัติ ดังนี้

1. Consistency หมายถึง ทุกโหนดหรือทุกเซิร์ฟเวอร์ที่ใช้ในการเก็บข้อมูลในระบบฐานข้อมูลแบบไม่สัมพันธ์จะต้องมองเห็นข้อมูลที่เหมือนกันและสามารถเห็นข้อมูลในเวลาเดียวกัน

2. Availability หมายถึง ทุกการเรียกใช้งานระบบฐานข้อมูล ระบบฐานข้อมูลจะต้องสามารถตอบกลับการเรียกใช้งานได้ทุกครั้ง ไม่ว่าระบบจะทำงานอยู่หรือมีปัญหาอยู่ก็ตาม

3. Partition Tolerance หมายถึง ระบบสามารถทำงานต่อไปได้แม้มีโหนดบางตัวหรือเซิร์ฟเวอร์บางตัวที่หยุดการทำงานลง

โดยคุณสมบัติของระบบสารสนเทศที่สามารถประมวลผลแบบกระจายที่ใช้งานในปัจจุบันไม่มีระบบชนิดใดเลยสามารถรองรับคุณสมบัติของทฤษฎี CAP ได้ทั้ง 3 คุณสมบัติ โดยระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละชนิดและแต่ละยี่ห้อ สามารถรองรับคุณสมบัติของทฤษฎี CAP ได้เพียงสองในสามคุณสมบัติเท่านั้น ซึ่งในรูปที่ 2.8 จะแสดงถึงคุณสมบัติของระบบฐานข้อมูลแต่ละชนิดที่รองรับทฤษฎี CAP



รูปที่ 2.8 คุณสมบัติของระบบฐานข้อมูลแต่ละชนิดที่รองรับทฤษฎี CAP [23]

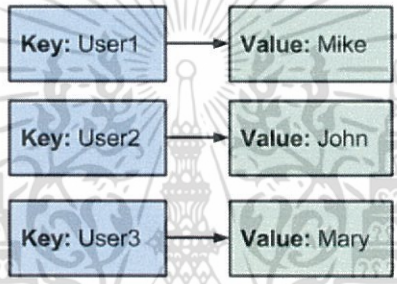
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.2 ประเภทของระบบฐานข้อมูลแบบไม่สัมพันธ์

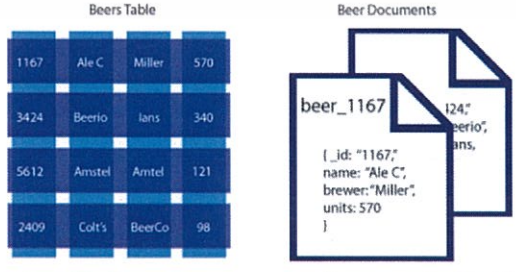
ระบบฐานข้อมูลแบบไม่สัมพันธ์มีรูปแบบการเก็บข้อมูลและสามารถแบ่งออกเป็น 4 ประเภท ดังนี้ [13]

1. ฐานข้อมูลแบบใช้คีย์อ้างอิง (Key Value Database) ทำการเก็บข้อมูลในลักษณะของคีย์และผลลัพธ์ โดยคีย์แต่ละตัวจะต้องเป็นค่าที่ไม่ซ้ำกับค่าของคีย์ตัวอื่น โดยการดึงเอาผลลัพธ์ออกมาจะอ้างอิงจากค่าของคีย์ ซึ่งฐานข้อมูลแบบใช้คีย์อ้างอิงเป็นฐานข้อมูลที่ไม่มีการกำหนดรูปแบบโครงสร้างเฉพาะ และลักษณะโครงสร้างแบบจำลองของฐานข้อมูลมีลักษณะอย่างง่ายและสามารถรองรับข้อมูลที่มีปริมาณมากได้ ยกตัวอย่างเช่นฐานข้อมูลเรดิส (Redis)และฐานข้อมูลไรอาค (Riak) เป็นต้น ดังแสดงรูปแบบการเก็บข้อมูลตัวอย่างในรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง [22]

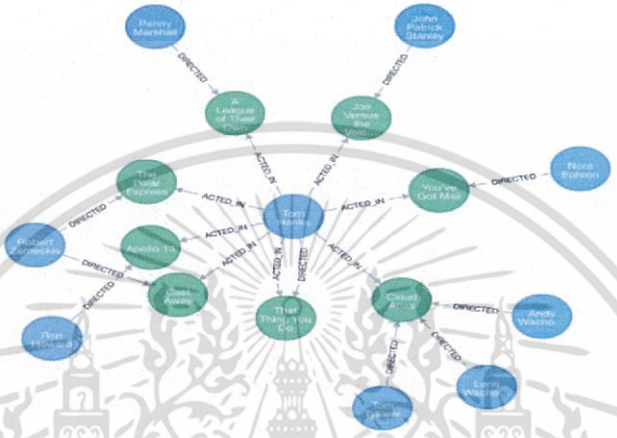
2. ฐานข้อมูลแบบเอกสาร (Document Database) ทำการเก็บข้อมูลในรูปแบบเอกสารและมีความยืนหยุ่นในการจัดเก็บข้อมูลและมีโครงสร้างทั้งแนวลึกและแนวกว้างในแต่ละแถว เหมือนกับอาเรย์หลายมิติแบบไม่แบนราบ ซึ่งแตกต่างจากระบบโครงสร้างแบบตารางของระบบฐานข้อมูลแบบสัมพันธ์ โดยมีหลากหลายรูปแบบ อาทิเช่น เจซันฟอร์แมต ไบนารี่เจสันฟอร์แมต เอ็กเอ็มแอลฟอร์แมต ไบนารี่ลาจออบเจ็คฟอร์แมต เป็นต้น ยกตัวอย่างเช่น ฐานข้อมูลมองโกดีบี (MongoDB)และฐานข้อมูลเค้าดีบี CouchDB เป็นต้น ดังรูปแบบการเก็บข้อมูลตัวอย่างในรูปที่ 2.10



รูปที่ 2.10 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสาร [24]

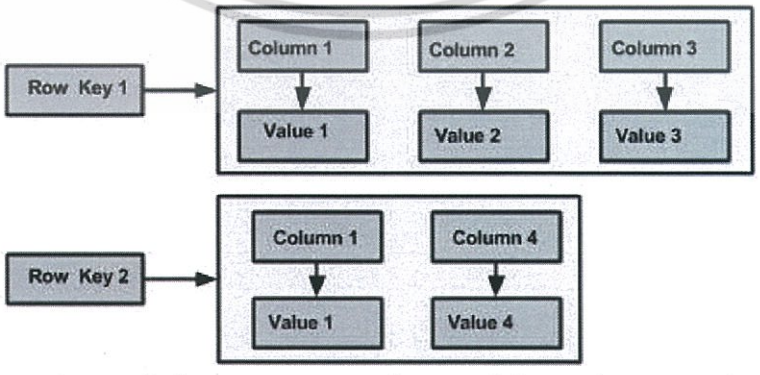
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
148682

3. ฐานข้อมูลแบบกราฟ (Graph Database) โดยข้อมูลจะถูกจัดเก็บในลักษณะของความสัมพันธ์ระหว่างโหนดหรือออบเจ็ค และข้อมูลจะถูกจัดเก็บในลักษณะของความสัมพันธ์แบบกราฟ ซึ่งจะมีโหนดเป็นตัวแทนของข้อมูลเชื่อมความสัมพันธ์แบบกราฟความสัมพันธ์ โดยสามารถรองรับการอ่านข้อมูลแบบซับซ้อนได้ดี ยกตัวอย่างเช่นฐานข้อมูลนีโอโฟร์เจ (Neo4J)และฐานข้อมูลโอเรี้ยนดีบี (OrientedDB) เป็นต้น ดังแสดงรูปแบบการเก็บข้อมูลตัวอย่างในรูปที่ 2.11



รูปที่ 2.11 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดกราฟ [25]

4. ฐานข้อมูลแบบคอลัมน์ขนาดใหญ่ (Column Oriented Database) เป็นฐานข้อมูลที่มีลักษณะคล้ายฐานข้อมูลเชิงสัมพันธ์ โดยฐานข้อมูลชนิดนี้จะจัดเก็บข้อมูลเป็นคอลัมน์หรือคอลัมน์ขนาดกว้าง และแต่ละคอลัมน์จะเก็บข้อมูลในรูปแบบคอลัมน์แฟมิลี่ ซึ่งแต่ละคอลัมน์แฟมิลี่จะรวบรวมเป็นแถว ในทางทฤษฎีแล้วฐานข้อมูลประเภทนี้สามารถเก็บข้อมูลที่มีขนาดของแถวได้ยาวแบบไม่จำกัด ยกตัวอย่างเช่น ฐานข้อมูลคาสสานดร้า (Cassandra) และฐานข้อมูลเอชเบส (Hbase) เป็นต้น ดังแสดงรูปแบบการเก็บข้อมูลตัวอย่างในรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดคอลัมน์ขนาดใหญ่ [26]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การเปรียบเทียบตัวอย่างระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละประเภท

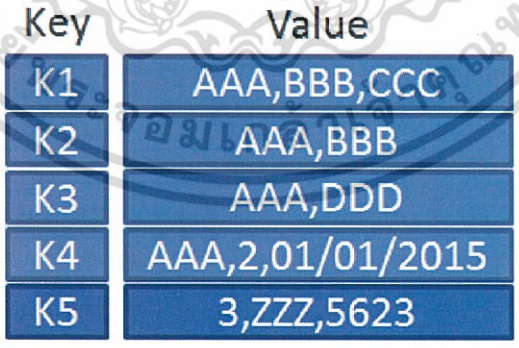
ตัวอย่างระบบฐานข้อมูล	ประเภทระบบฐานข้อมูล	คุณสมบัติ
MongoDB	ฐานข้อมูลแบบเอกสาร	เก็บข้อมูลในลักษณะไบนารี่เจสันฟอร์แมต และสามารถออกแบบโครงสร้างลักษณะคล้ายกับฐานข้อมูลเชิงสัมพันธ์ รองรับลักษณะ Consistency และ Partition Tolerance และรองรับรูปแบบการเก็บข้อมูลแบบไดนามิก (Dynamic) โดยสามารถกำหนดการเก็บข้อมูลรูปแบบชุดเอกสารย่อย (Sub Document) ได้ [13,23]
HBASE	ฐานข้อมูลแบบคอลัมน์ขนาดใหญ่	ถูกพัฒนามาจากฐานข้อมูลตารางขนาดใหญ่ (Big Table) โดยมี HDFS เป็นรากฐานในการเก็บข้อมูลเป็นหลัก โดยรองรับการขยายตัวได้ดี มีการใช้งานร่วมกับ Hadoop เหมาะสำหรับการใช้เป็นระบบฐานข้อมูลแบบกระจายและเหมาะสมกับการจัดการข้อมูลขนาดใหญ่ รองรับลักษณะ Consistency และ Partition Tolerance [13,23]
CouchDB	ฐานข้อมูลแบบเอกสาร	สามารถเขียนคำสั่งเรียกใช้งาน โดยการทำ MapReduce ผ่านจาวาสคริป โดยฐานข้อมูล CouchDB จะใช้งานผ่านเจสัน เอพีไอ (JSON API) ซึ่งทำให้สามารถเรียกได้จากทุกที่โดยผ่าน HTTP โพรโทคอลรองรับคุณสมบัติ Availability และ Partition Tolerance มีข้อจำกัดหลายอย่างเช่น ไม่เหมาะสมสำหรับการอ่านและเขียนข้อมูลพร้อมกัน [13,23]
Redis	ฐานข้อมูลแบบใช้คีย์อ้างอิง	เป็นฐานข้อมูลที่อยู่บนหน่วยความจำเป็นหลัก แต่ก็สามารถบันทึกข้อมูลลงบน ฮาร์ดดิสก์ ได้เช่นกัน รองรับการเก็บข้อมูลหลากหลายรูปแบบ อาทิเช่น List Sorted Sets Hashes Set String เป็นต้น และรองรับคุณสมบัติ Consistency และ Partition Tolerance [13,23] และเก็บข้อมูลที่มีปริมาณมากได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การเปรียบเทียบตัวอย่างระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละประเภท(ต่อ)

ตัวอย่างระบบฐานข้อมูล	ประเภทระบบฐานข้อมูล	คุณสมบัติ
Cassandra	ฐานข้อมูลแบบคอลัมน์ขนาดใหญ่	เป็นฐานข้อมูลที่รองรับการประมวลผลแบบกระจายได้ดี และใช้สถาปัตยกรรมแบบวงแหวน (Ring) เป็นต้นแบบในการทำการประมวลผลแบบกระจาย รองรับคุณสมบัติ Availability และ Partition Tolerance มีภาษาที่ใช้ในการเรียกใช้ข้อมูลเป็นของตนเอง (CQL : Cassandra Query Language) [13,23]
Neo4j	ฐานข้อมูลแบบกราฟ	ข้อมูลจะถูกจัดเก็บในลักษณะของความสัมพันธ์ระหว่างโหนดหรือออบเจ็ค และข้อมูลจะถูกจัดเก็บในลักษณะของความสัมพันธ์แบบกราฟ ซึ่งจะมีโหนดเป็นตัวแทนของข้อมูลเชื่อมความสัมพันธ์แบบกราฟความสัมพันธ์ โดยสามารถรองรับการอ่านข้อมูลแบบซับซ้อนได้ดี รองรับคุณสมบัติของ ACID ใช้ภาษา Cypher Query ในการเรียกอ่านข้อมูล [10,23]

2.2.3 ระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง



รูปที่ 2.13 รูปแบบการเก็บข้อมูลของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง[21]

ระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง [33] เป็นระบบฐานข้อมูลชนิดหนึ่งที่ถูกออกแบบมารองรับการเก็บข้อมูลและจัดการข้อมูลและการเรียกดูข้อมูล โดยมีโครงสร้างลักษณะการเก็บข้อมูลคล้ายกับอาร์เรย์ที่มีการเชื่อมโยง ซึ่งวิธีการใช้งานระบบฐานข้อมูลชนิดนี้มีลักษณะคล้ายการหาคำใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พจนานุกรม ซึ่งค่าข้อมูลผลลัพธ์ (Value) ที่ต้องการเก็บหรือต้องการเรียกดูสามารถอ้างอิงได้โดยค่าคีย์ของข้อมูล โดยข้อมูลผลลัพธ์ประกอบด้วยกลุ่มของข้อมูลหรือกลุ่มของออบเจ็ค และข้อมูลผลลัพธ์สามารถเก็บข้อมูลได้หลากหลายรูปแบบ อาทิเช่น สตริง (Strings) เจเพ็ก(JPEG) เจสัน (JSON) บิต (Bit) และตัวเลข (number) เป็นต้น โดยระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงเป็นรูปแบบการเก็บข้อมูลรูปแบบหนึ่งของระบบฐานข้อมูลแบบไม่สัมพันธ์ โดยมีเอกลักษณ์คือการใช้ค่าของคีย์ข้อมูลอ้างอิงค่าข้อมูลผลลัพธ์ ซึ่งเป็นรูปแบบการเก็บข้อมูลที่ง่ายและไม่ซับซ้อน และสามารถเรียกดูค่าข้อมูลผลลัพธ์ได้อย่างรวดเร็ว ซึ่งรูปแบบการเก็บข้อมูลของระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงสามารถแสดงให้เห็นดังรูปที่ 2.13

ระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงมีโครงสร้างและกระบวนการทำงานที่แตกต่างจากระบบฐานข้อมูลเชิงสัมพันธ์ เนื่องจากระบบฐานข้อมูลเชิงสัมพันธ์ต้องมีการกำหนดโครงสร้างและกำหนดความสัมพันธ์ของระบบฐานข้อมูลก่อนเสมอและต้องกำหนดชนิดของข้อมูลรวมทั้งคุณสมบัติของตารางข้อมูล โดยระบบฐานข้อมูลเชิงสัมพันธ์มีลักษณะโครงสร้างเป็นตารางข้อมูล 2 มิติ ประกอบด้วยแถวและสดมภ์ ซึ่งแตกต่างจากการเก็บข้อมูลของระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง เนื่องจากระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงทำการเก็บข้อมูลโดยอ้างอิงจากค่าของคีย์เพียงค่าเดียว ซึ่งระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงนั้นเป็นระบบฐานข้อมูลที่มีความยืดหยุ่นมากและใกล้เคียงกับแนวคิดสมัยใหม่ เช่นการประยุกต์ใช้กับการเขียนโปรแกรมเชิงวัตถุ และจากการวัดประสิทธิภาพการทำงานของระบบฐานข้อมูลทั้ง 2 ชนิด พบว่าระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงใช้พื้นที่หน่วยความจำน้อยกว่าระบบฐานข้อมูลเชิงสัมพันธ์ ทำให้สามารถเพิ่มประสิทธิภาพการทำงานระบบสารสนเทศได้ในอนาคต ซึ่งในอดีตระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงยังไม่เป็นที่นิยมมากเท่าที่ควร ทำให้มาตรฐานต่าง ๆ และการสนับสนุนหรือการพัฒนายังมีน้อยมาก แต่จากการพัฒนาเทคโนโลยีคลาวด์ (Cloud Technology) และการเพิ่มขึ้นของปริมาณข้อมูลสารสนเทศในปี 2010 ได้นำไปสู่ยุคที่เทคโนโลยีระบบฐานข้อมูลแบบไม่สัมพันธ์เฟื่องฟูมากขึ้น ทำให้ระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงถูกนำไปปรับใช้และนำไปพัฒนามากยิ่งขึ้น ซึ่งกระบวนการใช้คีย์อ้างอิงข้อมูลผลลัพธ์ยังเป็นรากฐานสำคัญให้กับระบบฐานข้อมูลชนิดกราฟอีกด้วย เพียงแต่ระบบฐานข้อมูลชนิดกราฟมีการสร้างความสัมพันธ์ระหว่างข้อมูลด้วย

โดยระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงสามารถจำแนกออกเป็น 4 ประเภท [33] ได้แก่

1. Key Value – eventually consistent : รองรับแบบจำลองความสอดคล้อง (Consistency Models) ยกตัวอย่างเช่น Dynamo, Oracle NoSQL Database, Project Voldermort, Riak และOpenLink Virtuoso เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Key Value – Ordered : รองรับการเรียงลำดับค่าของคีย์อ้างอิง ยกตัวอย่างเช่น Berkeley, FairCom c-treeACE/c-treeRTG, FoundationDB, HyperDex, IBM Informix C-ISAM และ MemcacheDB เป็นต้น

3. Key Value – Ram : รองรับการทำงานบนหน่วยความจำหลัก ยกตัวอย่างเช่น Aerospike, Coherence, FairCom c-treeACE, Redis, Memcached และ OpenLink Virtuoso เป็นต้น

4. Key Value – Solid-state drive or rotating disk : รองรับการจัดการระบบบน Solid-state drivers หรือ rotating disks ยกตัวอย่างเช่น Aerospike, CDB, Clusterpoint Database Server, Couchbase Server, LevelDB และKeyspace เป็นต้น

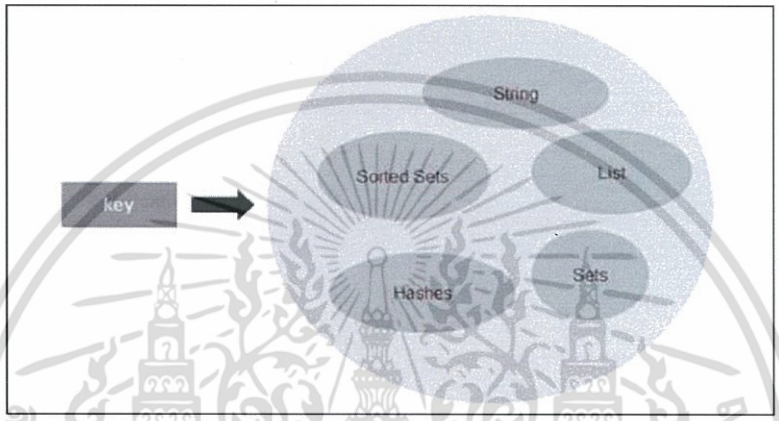
จากการสำรวจของ DB-Engines Ranking [20] ในปี 2016 พบว่าระบบฐานข้อมูลชนิดใช้คีย์อ้างอิงที่ได้รับความนิยมสูงสุด คือ เรดิส และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงยังสามารถนำไปใช้จัดการกับข้อมูลที่มีความหลากหลายได้ อาทิเช่น ข้อมูลเว็บเซสชัน (Session Information) ข้อมูลตะกร้าสินค้าบนเว็บไซต์ (Shopping Cart Contents) ข้อมูลรายละเอียดสินค้า (Product Details Data) ข้อมูลผู้ใช้ส่วนตัว (User Profiles Data) และ ข้อมูลสมุดโทรศัพท์ (Telecom Directories Data) เป็นต้น [30] ซึ่งฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงถือว่าได้รับความนิยมกับอย่างแพร่หลายในระบบสารสนเทศในปัจจุบัน ซึ่งมีแอปพลิเคชันที่มีชื่อเสียงมากมายเลือกใช้งานฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง อาทิเช่น กิตฮับ (GitHub) ทวิตเตอร์ (Twitter) เสต็ก โอเวอร์โฟล์ (Stack Overflow) พินเทอเรส (Pinterest) และ สแนปแชท (Snapchat) เป็นต้น ซึ่งถือได้ว่าฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงนั้น ได้รับความนิยมในการนำไปจัดการกับข้อมูลขนาดใหญ่

2.2.4 ฐานข้อมูลเรดิส (Redis)

ฐานข้อมูลเรดิส [34] คือฐานข้อมูลที่จัดอยู่ในกลุ่มโอเพนซอร์ซและจัดเป็นฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงชนิดหนึ่ง โดยฐานข้อมูลเรดิส นั้นทำการเก็บข้อมูลลงบนหน่วยความจำเป็นหลักหรือเรียกได้ว่าเป็นฐานข้อมูลแบบภายในหน่วยความจำ (In-Memory Data Structure) โดยฐานข้อมูลเรดิส สามารถไปประยุกต์ใช้เป็นระบบฐานข้อมูลหลักหรือระบบฐานข้อมูลชั่วคราว เป็นต้น ถึงแม้ว่าฐานข้อมูลเรดิส อาศัยการเก็บข้อมูลลงบนหน่วยความจำเป็นหลักแต่ฐานข้อมูลเรดิส สามารถเก็บข้อมูลลงบนฮาร์ดดิสก์ได้อย่างถาวรตามความต้องการของผู้ใช้งาน และผู้ใช้สามารถกำหนดระยะเวลาของข้อมูลที่มีอยู่ในระบบฐานข้อมูลได้ เพราะฉะนั้นข้อมูลจะไม่หายไปเมื่อทำการเริ่มทำงานใหม่ของระบบ (Restart) โดยฐานข้อมูลเรดิส มีรูปแบบหรือโครงสร้างที่ใช้ในการจัดเก็บข้อมูลอยู่หลายลักษณะ อาทิเช่น Strings Hashes Lists SetsและSorted Sets เป็นต้น โดยแต่ละรูปแบบมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะรูปแบบการใช้งานที่แตกต่างกันออกไป โดยขึ้นอยู่กับผู้ใช้จะเลือกใช้รูปแบบการเก็บข้อมูลรูปแบบใด โดยข้อดีของฐานข้อมูลเรดิส เป็นฐานข้อมูลที่มีโครงสร้างในการเก็บข้อมูลที่ไม่ซับซ้อนและมีรูปแบบการเก็บข้อมูลที่หลากหลายรวมทั้งยังมีความยืดหยุ่นสูงและฐานข้อมูลเรดิส ยังเป็นฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่ได้รับความนิยมสูงสุด จากการสำรวจของ DB-Engines Ranking [20] ดังนั้นทางผู้จัดทำจึงได้เลือกใช้ฐานข้อมูลเรดิส ในงานวิทยานิพนธ์ฉบับนี้ และแต่ละรูปแบบที่ใช้ในการจัดเก็บข้อมูลในฐานข้อมูลเรดิส จะถูกอ้างอิงด้วยค่าของคีย์ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 การเข้าถึงข้อมูลแต่ละรูปแบบผ่านค่าของคีย์บนระบบฐานข้อมูลเรดิส [14]

โดยฐานข้อมูลเรดิส มีรูปแบบโครงสร้างในการจัดเก็บข้อมูลดังนี้ [15]

- String
- Hashes
- Lists
- Sets
- Sorted Sets

2.2.4.1 คีย์ของฐานข้อมูลเรดิส

คีย์ของฐานข้อมูลเรดิส มีลักษณะเป็นเลขฐานสอง (Binary Safe) ซึ่งมีความหมายว่าสามารถใช้ข้อมูลที่เป็นเลขฐานสองตั้งค่าเป็นคีย์ของข้อมูลได้ โดยการตั้งค่าคีย์ของข้อมูลมีกฎดังต่อไปนี้ [15]

- ไม่ควรตั้งคีย์ของข้อมูลยาวจนเกินไป เนื่องจากคีย์สามารถตั้งค่าได้ไม่เกิน 512 เมกะไบต์ โดยการตั้งคีย์ของข้อมูลยาวเกินไปจะทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูล
- ไม่ควรตั้งคีย์ของข้อมูลสั้นจนเกินไป เนื่องจากถ้าตั้งค่าคีย์ของข้อมูลสั้นจนเกินไปอาจเกินความสับสนและไม่สามารถตีความหมายของคีย์นั้นออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ควรมีการใช้เครื่องหมายเพื่อช่วยให้การตั้งค่าคีย์ของข้อมูล ยกตัวอย่างเช่น “Object-type:id” จากตัวอย่างสามารถใช้เครื่องหมายทวิภาค (:) หรือเครื่องหมายมหัพภาค (.) เพื่อช่วยให้สามารถทำความเข้าใจค่าและความหมายของคีย์ได้ง่ายขึ้น

2.2.4.2 รูปแบบการเก็บข้อมูลแบบ Strings

```
> set mykey somevalue
OK
> get mykey
"somevalue"
```

รูปที่ 2.15 ตัวอย่างการใช้คำสั่ง SET และGET [15]

Strings [15] เป็นรูปแบบอย่างง่ายที่สุดในการเก็บข้อมูลของฐานข้อมูลเรดิส ซึ่งสามารถใช้คีย์เป็นรูปแบบ Strings อ้างอิงถึงค่าผลลัพธ์ ที่มีรูปแบบ Strings ได้ โดยมีคำสั่ง SET และ GET ในการบันทึกข้อมูลและอ่านข้อมูลออกมา โดยค่าผลลัพธ์ สามารถบันทึกค่าข้อมูลได้หลากหลาย อาทิเช่น String integer float jpeg file เป็นต้น โดยจะถูกอ้างอิงโดยคีย์ที่มีรูปแบบเป็น Strings โดยการใช้คำสั่ง SET ในการเขียนข้อมูล แต่การเขียนข้อมูลจะไม่สำเร็จ ถ้ามีคีย์ที่ได้ทำการบันทึกไปแล้ว หรืออีกความหมายหนึ่งคือไม่สามารถบันทึกคีย์ซ้ำได้ โดยรูปที่ 2.15 แสดงตัวอย่างการใช้คำสั่ง SET และ GET

```
> set counter 100
OK
> incr counter
(integer) 101
> incr counter
(integer) 102
> incrby counter 50
(integer) 152
```

รูปที่ 2.16 ตัวอย่างการใช้คำสั่ง INCR [15]

โดยฐานข้อมูลเรดิส มีคำสั่งที่สามารถเพิ่มหรือลดค่าของผลลัพธ์ที่มีลักษณะเป็น integer ได้ จากคำสั่ง INCR DECR และ DECRBY ดังแสดงตัวอย่างในรูปที่ 2.16 และฐานข้อมูลเรดิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแก้ไขค่าของผลลัพธ์ ได้จากการใช้คำสั่ง GETSET ซึ่งคำสั่งนี้จะแสดงค่าผลลัพธ์ เดิมออกมา จากนั้นจึงสามารถใส่ค่าผลลัพธ์ใหม่ลงไปได้

```
> mset a 10 b 20 c 30
OK
> mget a b c
1) "10"
2) "20"
3) "30"
```

รูปที่ 2.17 ตัวอย่างแสดงการใช้คำสั่ง MSET และ MGET [15]

ฐานข้อมูลเรดิส สามารถใช้คำสั่ง MSET และ MGET ในการบันทึกข้อมูลและอ่านข้อมูลออกมาได้มากกว่า 1 คีย์ โดยคำสั่ง MGET จะคืนค่าเป็นอาร์เรย์ของข้อมูลผลลัพธ์ออกมา ดังแสดงตัวอย่างในรูปที่ 2.17 และฐานข้อมูลเรดิส ยังสามารถปรับเปลี่ยนหรือตรวจสอบค่าของคีย์ที่ถูกบันทึกในฐานข้อมูลเรดิส ผ่านคำสั่ง EXISTS โดยเมื่อใช้คำสั่งนี้ หากผลลัพธ์คืนค่า 1 แสดงว่ามีคีย์ที่อ้างอิงได้ถูกบันทึกในระบบฐานข้อมูลแล้ว แต่หากผลลัพธ์คืนค่า 0 แสดงว่าไม่มีคีย์นี้อยู่ในระบบฐานข้อมูล และระบบฐานข้อมูลเรดิส สามารถลบคีย์ได้ผ่านคำสั่ง DEL และฐานข้อมูล เรดิส สามารถตรวจสอบประเภทของข้อมูลที่ใช้ในการกำหนดค่าของคีย์ข้อมูลผ่านคำสั่ง TYPE ดังแสดงตัวอย่างในรูปที่ 2.18

```
> set mykey hello
OK
> exists mykey
(integer) 1
> del mykey
(integer) 1
> exists mykey
(integer) 0
```

```
> set mykey x
OK
> type mykey
string
> del mykey
(integer) 1
> type mykey
none
```

รูปที่ 2.18 ตัวอย่างแสดงการใช้คำสั่ง EXISTS และ DEL [15]

ฐานข้อมูลเรดิส ยังสามารถกำหนดระยะเวลาของคีย์ที่ถูกบันทึกในระบบฐานข้อมูลได้ ผ่านคำสั่ง EXPIRE หรือเรียกอีกอย่างว่ากำหนด Time to live โดยค่าปกติของคำสั่ง EXPIRE คือ 1 มิลลิวินาที ดังแสดงในเห็นในตัวอย่างจากรูปที่ 2.19

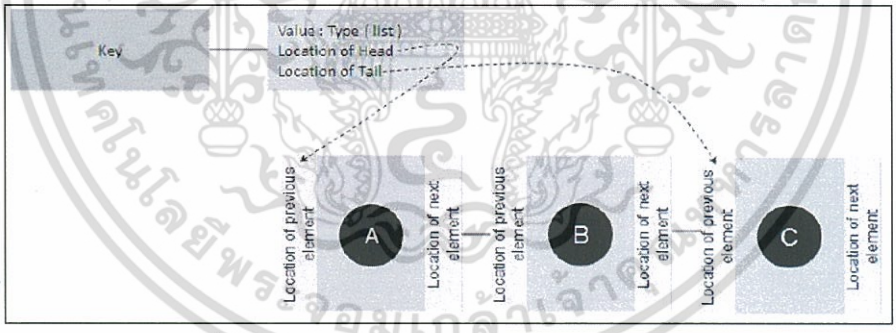
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
> set key some-value
OK
> expire key 5
(integer) 1
> get key (immediately)
"some-value"
> get key (after some time)
(nil)
```

รูปที่ 2.19 ตัวอย่างการใช้คำสั่ง EXPIRE [15]

2.2.4.3 รูปแบบการเก็บข้อมูลแบบ List

List [15] เป็นรูปแบบการเก็บข้อมูลที่มีลักษณะเปรียบเสมือนกลุ่มของ Strings โดยรูปแบบการเก็บข้อมูลแบบ List นั้นสามารถเรียงลำดับข้อมูลตามลำดับการบันทึกข้อมูลได้ โดยสามารถบันทึกข้อมูลลงไปทางหัวของ List หรือบันทึกต่อท้ายของ List โดยรูปแบบการเก็บข้อมูลแบบ List มีลักษณะคล้าย Linked list เพราะฐานข้อมูลเรดิส สามารถบันทึกข้อมูลในรูปแบบของ List ขนาดยาวได้ ซึ่งแสดงให้เห็นดังรูปที่ 2.20



รูปที่ 2.20 ภาพจำลองการเก็บข้อมูลแบบ List [14]

โดยรูปแบบการเก็บข้อมูลแบบ List สามารถบันทึกข้อมูลโดยผ่านคำสั่ง LPUSH โดยเป็นการบันทึกข้อมูลทางด้านซ้ายหรือบันทึกข้อมูลต่อตรงส่วนหัวของ List และคำสั่ง RPUSH โดยเป็นการบันทึกข้อมูลทางด้านขวาหรือบันทึกข้อมูลต่อตรงส่วนท้ายของ List และคำสั่ง LRANGE เป็นคำสั่งที่อ่านค่าข้อมูลที่ถูกบันทึกภายใน List โดยผู้ใช้สามารถอ่านข้อมูลโดยกำหนดลำดับของข้อมูลได้ ดังแสดงตัวอย่างในรูปที่ 2.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
> rpush mylist A
(integer) 1
> rpush mylist B
(integer) 2
> lpush mylist first
(integer) 3
> lrange mylist 0 -1
1) "first"
2) "A"
3) "B"
```

```
> rpush mylist 1 2 3 4 5 "foo ba
(integer) 9
> lrange mylist 0 -1
1) "first"
2) "A"
3) "B"
4) "1"
5) "2"
6) "3"
7) "4"
8) "5"
9) "foo bar"
```

รูปที่ 2.21 ตัวอย่างการใช้คำสั่ง LPUSH RPUSH และ LRANGE [15]

โดยรูปแบบการเก็บข้อมูลแบบ List สามารถดึงข้อมูลออกมาได้ผ่านคำสั่ง POP และการดึงข้อมูลออกมาจากรูปแบบการเก็บข้อมูลแบบ List สามารถทำได้โดยผ่านคำสั่ง LPOP และ RPOP โดยคำสั่ง LPOP เป็นการดึงข้อมูลตัวแรกจากทางซ้ายสุดหรือส่วนหัวของ List และคำสั่ง RPOP เป็นการดึงข้อมูลตัวแรกจากทางขวาสุดหรือส่วนท้ายของ List แสดงตัวอย่างในเห็นในรูปที่ 2.22

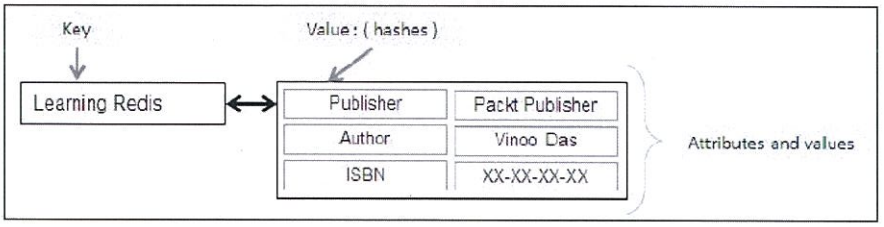
```
> rpush mylist a b c
(integer) 3
> rpop mylist
"c"
> rpop mylist
"b"
> rpop mylist
"a"
```

รูปที่ 2.22 ตัวอย่างการใช้คำสั่ง RPOP และ LPOP [15]

2.2.4.4 รูปแบบการเก็บข้อมูลแบบ Hashes

Hashes [15] เป็นรูปแบบการเก็บข้อมูลที่สามารถเก็บข้อมูลผลลัพธ์ และเก็บข้อมูลแอททริบิวต์ (Attribute) ร่วมกันได้ และเป็นรูปแบบการเก็บข้อมูลที่สามารถนำไปปรับใช้กับการเก็บข้อมูลที่มีรูปแบบเป็นออบเจ็คได้ดี คล้ายกับการเก็บข้อมูลในรูปแบบ JSON หรืออ้างอิงข้อมูลที่บันทึกในรูปแบบของตารางได้ ซึ่งแสดงให้เห็นภาพในรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 ภาพจำลองการเก็บข้อมูลแบบ Hashes [14]

โดยรูปแบบการเก็บข้อมูลแบบ Hashes มีคำสั่งในการบันทึกข้อมูลคือ HMSET ซึ่งคำสั่ง HMSET สามารถบันทึกข้อมูลผลลัพธ์และแอททริบิวต์ได้มากกว่าหนึ่งค่าข้อมูล และมีคำสั่ง HGET ในการดึงข้อมูลออกมาเพียงข้อมูลเดียว และมีคำสั่ง HMGET ในการดึงข้อมูลออกมาได้หลายข้อมูล ดังแสดงตัวอย่างในรูปที่ 2.24

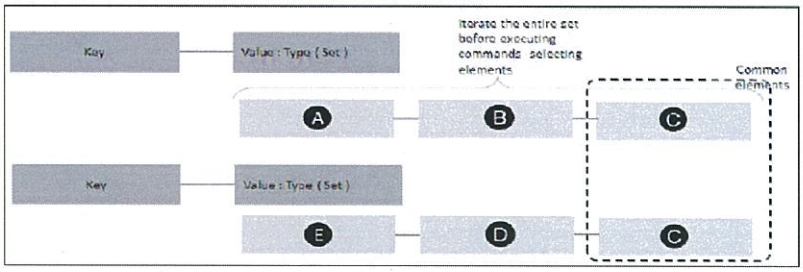
```
> hmset user:1000 username antirez birthyear 1977 verified 1
OK
> hget user:1000 username
"antirez"
> hget user:1000 birthyear
"1977"
> hgetall user:1000
1) "username"
2) "antirez"
3) "birthyear"
4) "1977"
5) "verified"
6) "1"
```

```
> hmget user:1000 username birthyear no-such-field
1) "antirez"
2) "1977"
3) (nil)
```

รูปที่ 2.24 ตัวอย่างการใช้คำสั่ง HMSET HGET และ HMGET [15]

2.2.4.5 รูปแบบการเก็บข้อมูลแบบ Sets

รูปแบบการเก็บข้อมูลแบบ Sets [15] เป็นการเก็บข้อมูลแบบกลุ่มของ Strings โดยรูปแบบการเก็บข้อมูลผลลัพธ์แบบ Sets จะไม่เก็บข้อมูลซ้ำและไม่เรียงลำดับข้อมูล และคุณลักษณะของการเก็บข้อมูลแบบ Sets สามารถรองรับการยูเนียน (Union) การอินเตอร์เซคชัน (Intersection) การหาค่าผลต่าง (Difference) ดังแสดงตัวอย่างในรูปที่ 2.25



รูปที่ 2.25 ภาพจำลองการเก็บข้อมูลแบบ Sets [14]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรูปแบบการเก็บข้อมูลแบบ Sets ใช้คำสั่ง SADD ในการบันทึกข้อมูลลงบน Sets และใช้คำสั่ง SMEMBERS ในการดูข้อมูลผลลัพธ์ที่เป็นสมาชิกผ่านใน Sets และใช้คำสั่ง SISMEMBER ในการตรวจสอบว่า ข้อมูลผลลัพธ์ตัวใดเป็นสมาชิกของ Sets นั้น ๆ หรือไม่ ดังแสดงตัวอย่างดังรูปที่ 2.26

```
> sadd myset 1 2 3
(integer) 3
> smembers myset
1. 3
2. 1
3. 2
```

```
> sismember myset 3
(integer) 1
> sismember myset 30
(integer) 0
```

รูปที่ 2.26 ตัวอย่างการใช้คำสั่ง SADD SMEMBERS และ SISMEMBER [15]

รูปแบบการเก็บข้อมูลแบบ Sets ยังสามารถรองรับคุณสมบัติทางคณิตศาสตร์ผ่านทางทฤษฎีของเซตได้ เช่น การยูเนียน การอินเตอร์เซคชัน การหาค่าผลต่าง เป็นต้น โดยการยูเนียนจะใช้งานผ่านคำสั่ง SUNION และการอินเตอร์เซคชันจะใช้งานผ่านคำสั่ง SINTER และการหาค่าผลต่าง จะใช้งานผ่านคำสั่ง SDIFF ดังแสดงตัวอย่างในรูปที่ 2.27

```
redis> SADD key1 "a"
(integer) 1
redis> SADD key1 "b"
(integer) 1
redis> SADD key1 "c"
(integer) 1
redis> SADD key2 "c"
(integer) 1
redis> SADD key2 "d"
(integer) 1
redis> SADD key2 "e"
(integer) 1
redis> SUNION key1 key2
1) "c"
2) "e"
3) "a"
4) "b"
5) "d"
redis>
```

```
redis> SADD key1 "a"
(integer) 1
redis> SADD key1 "b"
(integer) 1
redis> SADD key1 "c"
(integer) 1
redis> SADD key2 "c"
(integer) 1
redis> SADD key2 "d"
(integer) 1
redis> SADD key2 "e"
(integer) 1
redis> SINTER key1 key2
1) "c"
redis>
```

```
redis> SADD key1 "a"
(integer) 1
redis> SADD key1 "b"
(integer) 1
redis> SADD key1 "c"
(integer) 1
redis> SADD key2 "c"
(integer) 1
redis> SADD key2 "d"
(integer) 1
redis> SADD key2 "e"
(integer) 1
redis> SDIFF key1 key2
1) "a"
2) "b"
redis>
```

รูปที่ 2.27 ตัวอย่างการใช้คำสั่ง SUNION [18] SINTER [17] และ SDIFF [16]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรูปแบบการเก็บข้อมูลแบบ Sets มีคำสั่งที่ใช้ในการดึงข้อมูลออกมาจาก Sets โดยทำการสุ่มข้อมูลผลลัพธ์จาก Sets ซึ่งคำสั่งที่ใช้คือคำสั่ง SPOP ดังแสดงตัวอย่างให้เห็นในรูปที่ 2.28

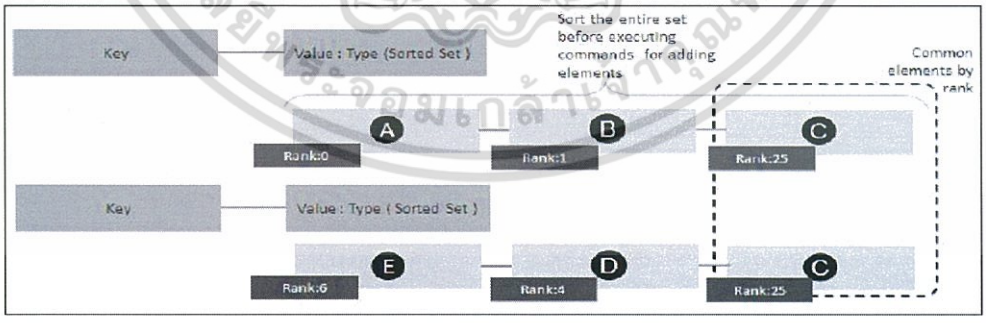
```
redis> SADD myset "one"
(integer) 1
redis> SADD myset "two"
(integer) 1
redis> SADD myset "three"
(integer) 1
redis> SPOP myset
"two"
redis> SMEMBERS myset
1) "three"
2) "one"
```

```
redis> SADD myset "four"
(integer) 1
redis> SADD myset "five"
(integer) 1
redis> SPOP myset 3
1) "three"
2) "four"
3) "one"
redis> SMEMBERS myset
1) "five"
```

รูปที่ 2.28 ตัวอย่างการใช้คำสั่ง SPOP [19]

2.2.4.6 รูปแบบการเก็บข้อมูลแบบ Sorted Sets

รูปแบบการเก็บข้อมูลแบบ Sorted Sets [15] เป็นรูปแบบการเก็บข้อมูลที่เปรียบเสมือนการรวมกันระหว่างรูปแบบการเก็บข้อมูลแบบ Sets และ Hashes โดยข้อมูลผลลัพธ์ของรูปแบบการเก็บข้อมูลแบบ Sorted Sets จะเก็บข้อมูลเป็นกลุ่มข้อมูลและไม่มีการเรียงลำดับข้อมูล แต่ข้อมูลผลลัพธ์จะแปรผันหรืออ้างอิงตามค่าของคะแนน (Score) ซึ่งค่าของคะแนนจะจับคู่กับค่าของข้อมูลผลลัพธ์ คล้ายการเก็บข้อมูลแบบ Hashes ซึ่งแสดงตัวอย่างในรูปที่ 2.29



รูปที่ 2.29 ภาพจำลองการเก็บข้อมูลแบบ Sorted Sets [14]

โดยรูปแบบการเก็บข้อมูลแบบ Sorted Sets ใช้คำสั่ง ZADD ในการบันทึกข้อมูล คล้ายการการใช้งานคำสั่ง SADD แต่คำสั่ง ZADD จะต้องมีการกำหนดค่าคะแนนของข้อมูลด้วย ดังแสดงตัวอย่างในรูปที่ 2.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
> zadd hackers 1940 "Alan Kay"
(integer) 1
> zadd hackers 1957 "Sophie Wilson"
(integer) 1
> zadd hackers 1953 "Richard Stallman"
(integer) 1
> zadd hackers 1949 "Anita Borg"
(integer) 1
> zadd hackers 1965 "Yukihiro Matsumoto"
(integer) 1
> zadd hackers 1914 "Hedy Lamarr"
(integer) 1
> zadd hackers 1916 "Claude Shannon"
(integer) 1
> zadd hackers 1969 "Linus Torvalds"
(integer) 1
> zadd hackers 1912 "Alan Turing"
(integer) 1
```

รูปที่ 2.30 ตัวอย่างการใช้คำสั่ง ZADD [15]

โดยรูปแบบการเก็บข้อมูลแบบ Sorted Sets ใช้คำสั่ง ZRANGE และ ZREVRANGE ในการดึงข้อมูลผลลัพธ์ออกมา ซึ่งคำสั่ง ZRANGE จะดึงข้อมูลผลลัพธ์โดยเรียงลำดับจากค่าคะแนนของข้อมูลน้อยไปมาก และในทางตรงกันข้าคำสั่ง ZREVRANGE จะดึงข้อมูลผลลัพธ์โดยเรียงลำดับจากค่าคะแนนของข้อมูลมากไปน้อย ดังแสดงตัวอย่างในรูปที่ 2.31

```
> zrange hackers 0 -1
1) "Alan Turing"
2) "Hedy Lamarr"
3) "Claude Shannon"
4) "Alan Kay"
5) "Anita Borg"
6) "Richard Stallman"
7) "Sophie Wilson"
8) "Yukihiro Matsumoto"
9) "Linus Torvalds"
```

```
> zrevrange hackers 0 -1
1) "Linus Torvalds"
2) "Yukihiro Matsumoto"
3) "Sophie Wilson"
4) "Richard Stallman"
5) "Anita Borg"
6) "Alan Kay"
7) "Claude Shannon"
8) "Hedy Lamarr"
9) "Alan Turing"
```

รูปที่ 2.31 ตัวอย่างการใช้คำสั่ง ZRANGE และ ZREVRANGE [15]

2.3 งานวิจัยที่เกี่ยวข้อง

ทางผู้จัดทำวิทยานิพนธ์ได้ทำการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง ที่สามารถนำมาประยุกต์ใช้ในงานวิจัยได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการศึกษางานวิจัยที่เกี่ยวข้องกับระบบฐานข้อมูลพบว่ามีบทความและงานวิจัยมากมายที่นำเสนอถึงการเปรียบเทียบระหว่างระบบฐานข้อมูลแบบไม่สัมพันธ์และระบบฐานข้อมูลเชิงสัมพันธ์ เนื่องจากนักวิจัยเหล่านี้ต้องการหาระบบฐานข้อมูลที่เหมาะสมกับระบบแอปพลิเคชันที่ต้องการ

ซึ่งมีงานวิจัยจำนวนมากที่นำเสนอการเปรียบเทียบประสิทธิภาพการประมวลผลระหว่างฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารกับฐานข้อมูลเชิงสัมพันธ์ ยกตัวอย่างเช่น มีงานวิจัยที่นำฐานข้อมูลมองโกดีบี มาเปรียบเทียบกับฐานข้อมูลเอสคิวแอล เซิร์ฟเวอร์ อาทิเช่น งานวิจัย [1] นักเขียนได้ทำการทดลองอยู่บนโปรแกรมภาษาซีชาร์ป (C#) แอปพลิเคชัน โดยใช้โครงสร้างฐานข้อมูล 3 ตาราง ได้แก่ ตาราง department ตาราง user ตาราง project และแบ่งขนาดของชุดข้อมูลออกเป็น 4 ระดับ (Test Case) โดยทำการทดลองเขียนข้อมูลลงบนฐานข้อมูลทั้ง 2 ชนิด และทดลองใน 4 ระดับขนาดข้อมูลสำหรับข้อมูลทั้ง 3 ตาราง และยังทำการทดลองในส่วนของการแก้ไขข้อมูล ซึ่งการทดลองในส่วนของการแก้ไขข้อมูลถูกแบ่งออกเป็น 3 รูปแบบ และผู้เขียนยังทำการทดลองการอ่านข้อมูลหลากหลายรูปแบบ ทั้งการอ่านข้อมูลอย่างง่ายและการอ่านข้อมูลแบบซับซ้อน เมื่อทำการเปรียบเทียบประสิทธิภาพระหว่าง ฐานข้อมูลมองโกดีบี และฐานข้อมูลเอสคิวแอล เซิร์ฟเวอร์ ใน 3 การทดลองนี้ ผลการทดลองได้แสดงให้เห็นว่าฐานข้อมูล มองโกดีบี มีประสิทธิภาพในการเขียน แก้ไข และ อ่านข้อมูลแบบอย่างง่ายดีกว่าฐานข้อมูลเอสคิวแอล เซิร์ฟเวอร์ แต่ฐานข้อมูลเอสคิวแอล เซิร์ฟเวอร์ จะมีประสิทธิภาพที่ดีกว่าฐานข้อมูลมองโกดีบี ในการทดลองแก้ไขข้อมูลแบบ ไม่ใช้คีย์หลักอ้างอิง และอ่านข้อมูลแบบไม่ใช้คีย์หลักอ้างอิง และการอ่านข้อมูลแบบซับซ้อนชนิด ฟังก์ชันการรวม (Aggregate Function) และงานวิจัย [9] นักเขียนได้เปลี่ยนแปลงรูปแบบการทดลอง รวมถึงเปลี่ยนแปลงขนาดของชุดข้อมูลที่นำมาทำการทดลอง โดยทำการทดลองเปรียบเทียบบนรากฐานของโครงสร้างแบบจำลองของอี-คอมเมิร์ซ ซึ่งแบ่งการทดลองออกเป็น 5 รูปแบบ ประกอบด้วย การเขียน การแก้ไข การลบ การอ่านและการอ่านแบบครอบคลุม และงานวิจัยนี้ได้ทำการทดลองอ่านข้อมูลจากระบบฐานข้อมูลในหลากหลายรูปแบบ ประกอบด้วย 1. การอ่านข้อมูลโดยปราศจากการใช้ฟังก์ชันการรวม 2. การอ่านข้อมูลโดยใช้ฟังก์ชันการรวม ซึ่งผลการทดลองสามารถสรุปได้ว่าฐานข้อมูลมองโกดีบี มีประสิทธิภาพดีกว่าฐานข้อมูลเอสคิวแอล เซิร์ฟเวอร์ ในรูปแบบของการเขียน การแก้ไข การลบ และการอ่านข้อมูลแบบปราศจากการใช้ฟังก์ชันการรวม แต่ฐานข้อมูลมองโกดีบี จะมีประสิทธิภาพที่ด้อยกว่าฐานข้อมูล เอสคิวแอล เซิร์ฟเวอร์ ในโอเปอเรชั่นการอ่านข้อมูลแบบฟังก์ชันการรวมเป็นต้น

บางงานวิจัยนำฐานข้อมูล มองโกดีบี มาเปรียบเทียบกับฐานข้อมูลมายเอสคิวแอล อาทิเช่น [5] นี่คืองานวิจัยมีจุดประสงค์เพื่อพิสูจน์ว่าทำไมฐานข้อมูล มองโกดีบี จึงมีประสิทธิภาพที่ดีกว่าฐานข้อมูลมายเอสคิวแอล โดยทำการทดลองบนฟอรัมแอปพลิเคชัน ซึ่งแบ่งการทดลองออกเป็น 4 ตัวดำเนินการ ประกอบด้วยการเขียน การแก้ไข การลบและการอ่าน โดยผลการทดลองแสดงให้เห็นว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลมองโกดีบี มีประสิทธิภาพที่ดีกว่าฐานข้อมูลมายเอสคิวแอล และในงานวิจัย [7] ทำการทดลองของงานวิจัยชิ้นนี้ทดลองอยู่บนโครงสร้างฐานข้อมูลของสายการบินขนาด 1,050,000 ระเบียน ซึ่งเป็นแบบจำลองที่มีขนาดใหญ่ โดยงานวิจัยชุดนี้แบ่งการทดลองออกเป็น 4 ตัวดำเนินการ ประกอบด้วย การเขียน การอ่าน การแก้ไขและการลบ ซึ่งผลการทดลองที่ได้แสดงให้เห็นว่าฐานข้อมูลมองโกดีบี มีประสิทธิภาพดีกว่าฐานข้อมูลมายเอสคิวแอล ในแอปพลิเคชันที่มีโครงสร้างฐานข้อมูลขนาดใหญ่ ในงานวิจัย [2] นักเขียนต้องการพิสูจน์หาประสิทธิภาพการทำงานของระบบฐานข้อมูลจัดการหนังสือดิจิตอลบุ๊ค เมื่อ แทนที่ระบบฐานข้อมูลเชิงสัมพันธ์ด้วยระบบฐานข้อมูลแบบไม่สัมพันธ์ ซึ่งฐานข้อมูลแบบไม่สัมพันธ์ที่นำมาใช้ในการทดลอง คือ มองโกดีบี และระบบฐานข้อมูลเชิงสัมพันธ์ ที่นำมาใช้ในการทดลองคือ มายเอสคิวแอล ซึ่งผลการทดลองที่ได้คือฐานข้อมูล มองโกดีบี มีประสิทธิภาพที่ดีกว่าฐานข้อมูลมายเอสคิวแอล ในส่วนของการเขียนข้อมูลและอ่านข้อมูล เป็นต้น

บางงานวิจัยนำฐานข้อมูลมองโกดีบี มาเปรียบเทียบกับฐานข้อมูลออราเคิล อาทิเช่น งานวิจัย [3] นักเขียนได้นำเสนอถึงความแตกต่างขององค์ประกอบต่าง ๆ ของฐานข้อมูลทั้ง 2 ชนิด และเปรียบเทียบความเร็วในการประมวลผลด้านการเขียน ด้านการแก้ไขและด้านการลบ โดยทำการทดลองบนข้อมูล 3 สดมภ์ ระดับขนาดข้อมูลน้อยไปขนาดข้อมูลมาก ซึ่งผลการทดลองที่ได้นั้น ฐานข้อมูลมองโกดีบี มีประสิทธิภาพที่ดีกว่าฐานข้อมูลออราเคิล ในทุกการทดลอง และในงานวิจัย [8] นักเขียนต้องการทำการทดลองอ่านข้อมูลในรูปแบบที่ซับซ้อนจากข้อมูลที่มีขนาดใหญ่ ซึ่งจากผลการทดลองแสดงให้เห็นว่าฐานข้อมูล มองโกดีบี มีประสิทธิภาพที่ดีกว่าฐานข้อมูลออราเคิล ในการอ่านข้อมูลแบบไม่ซับซ้อน แต่ฐานข้อมูลออราเคิล จะมีประสิทธิภาพที่ดีกว่าฐานข้อมูลมองโกดีบี ในการอ่านข้อมูลที่มีรูปแบบซับซ้อนชนิดใช้ฟังก์ชันการรวม ซึ่งการอ่านข้อมูลแบบฟังก์ชันการรวม ประกอบด้วย การหาค่าเฉลี่ย การหาผลรวม การนับจำนวน

งานวิจัย [6] นักเขียนต้องการวิจัยและศึกษาเพื่อเปรียบเทียบประสิทธิภาพระหว่างระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารและระบบฐานข้อมูลเชิงสัมพันธ์ โดยระบบฐานข้อมูลแบบไม่สัมพันธ์ที่นำมาใช้ในการทดลองคือ มองโกดีบี และระบบฐานข้อมูลเชิงสัมพันธ์ที่นำมาใช้ในการทดลองคือ โพสต์เกรสคิวเอล ซึ่งใช้แบบจำลองโครงสร้างฐานข้อมูลของ Logical and physical ERD ซึ่งแบบจำลองโครงสร้างฐานข้อมูลประกอบด้วยตาราง user ตาราง user's card และตาราง mileage โดยทางผู้จัดทำได้แบ่งรูปแบบการทดลองออกเป็น 4 ตัวดำเนินการ ประกอบด้วย การเขียน การแก้ไข การลบ และการอ่าน ซึ่งทำการทดลองบนข้อมูลจำนวน 30,000 ข้อมูล 90,000 ข้อมูล 150,000 ข้อมูล 210,000 ข้อมูลและ 300,000 ข้อมูล ซึ่งบนฐานข้อมูลมองโกดีบี ได้ออกแบบแบบจำลองของโครงสร้างฐานข้อมูลเป็น 2 รูปแบบ คือ แบบจำลองเชิงสัมพันธ์และแบบจำลองแบบไม่สัมพันธ์ ซึ่งผลการทดลองสรุปได้ว่าฐานข้อมูล มองโกดีบี มีประสิทธิภาพในการทำงานดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลโพสต์เกรสคิวเอล โดยฐานข้อมูลมองโกดีบี ยังสามารถเพิ่มประสิทธิภาพให้สูงขึ้นได้ จากการใช้แบบจำลองแบบไม่สัมพันธ์ บางงานวิจัย [4] ต้องการเปรียบเทียบประสิทธิภาพของระบบฐานข้อมูลแบบไม่สัมพันธ์หลากหลายชนิดกับระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งระบบฐานข้อมูลแบบไม่สัมพันธ์ได้ถูกออกแบบเพื่อรองรับกับการเก็บข้อมูลแบบคู่คีย์อ้างอิงผลลัพธ์ ซึ่งตัวดำเนินการที่ใช้ในการทดลอง ประกอบด้วย การอินสแตนซ์ การอ่าน การเขียนและการลบ โดยระบบฐานข้อมูลแบบไม่สัมพันธ์ที่นำมาเปรียบเทียบประกอบด้วยระบบฐานข้อมูล มองโกดีบี เรเวียนดีบี (RavenDB) เค้าดีบี คาสสานดร้า ไฮเปอร์เทเบิล(HyperTable)และ เค้าเบส (Couchbase) และระบบฐานข้อมูลเชิงสัมพันธ์ที่นำมาเปรียบเทียบ คือ เอสคิวแอล เซิร์ฟเวอร์ ซึ่งผลลัพธ์ของการทดลองสรุปได้ว่าระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละชนิดมีผลการทดลองที่ดีและด้อยในแต่ละตัวดำเนินการที่แตกต่างกัน ซึ่งในแต่ละตัวดำเนินการไม่มีระบบฐานข้อมูลตัวไหนที่ดีที่สุด และฐานข้อมูลแบบไม่สัมพันธ์บางชนิดยังมีประสิทธิภาพที่แย่กว่าระบบฐานข้อมูลเชิงสัมพันธ์ในบางการทดลองอีกด้วย และงานวิจัย [10] นักเขียนต้องการทำวิจัยเพื่อเปรียบเทียบประสิทธิภาพระหว่างระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดกราฟ ที่มีชื่อว่านีโอโฟร์เจ และระบบฐานข้อมูลเชิงสัมพันธ์ที่มีชื่อว่า มายเอสคิวแอล และในงานวิจัยชิ้นนี้ยังเปรียบเทียบคุณลักษณะที่แตกต่างกันของระบบฐานข้อมูลทั้ง 2 ชนิด อาทิเช่น เปรียบเทียบด้านระดับการสนับสนุน ด้านการรักษาความปลอดภัยและด้านความยืดหยุ่น ซึ่งงานวิจัยชุดนี้ยังทำการทดลองวัดประสิทธิภาพการอ่านข้อมูลจากฐานข้อมูล ซึ่ง โครงสร้างฐานข้อมูลที่ใช้ในการทดลองนี้ประกอบด้วยข้อมูล 4 ตาราง คือ ตาราง student ตาราง friends ตาราง departments และตาราง faculty โดยผลการทดลองจากงานวิจัยชุดนี้ สามารถสรุปได้ว่าระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดกราฟมีประสิทธิภาพดีกว่าระบบฐานข้อมูลเชิงสัมพันธ์

จากงานวิจัยหลายฉบับที่ได้กล่าวมาจะเห็นว่า งานวิจัยส่วนใหญ่จะนำระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารมาใช้ในการเปรียบเทียบประสิทธิภาพกับระบบฐานข้อมูลเชิงสัมพันธ์เป็นจำนวนมาก โดย มองโกดีบี คือฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารที่ได้รับความนิยมสูงสุดในการนำมาเปรียบเทียบและนำมาทำงานวิจัย เนื่องจากฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารเป็นฐานข้อมูลที่มีประสิทธิภาพสูงและเป็นฐานข้อมูลแบบไม่สัมพันธ์ที่ได้รับความนิยมในวงการพัฒนาระบบสารสนเทศสูงสุด แต่จากผลการทดลองของหลายงานวิจัยจะพบว่าฐานข้อมูลมองโกดีบี ยังมีจุดด้อยหรือจุดอ่อนเมื่อเปรียบเทียบกับฐานข้อมูลเชิงสัมพันธ์ ดังนั้นทางผู้จัดทำจึงเล็งเห็นความสำคัญ ถึงระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดอื่นที่ความน่าสนใจ โดยในปัจจุบันระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เป็นระบบฐานข้อมูลที่ได้รับความนิยมไม่แพ้ระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดเอกสารและมีการนำฐานข้อมูลชนิดนี้ไปใช้งานกันอย่างแพร่หลายในระบบสารสนเทศในลักษณะต่าง ๆ ดังนั้นทางผู้จัดทำจึงเล็งเห็นถึงความสำคัญในระบบฐานข้อมูลชนิดใช้คีย์อ้างอิง และนำระบบฐานข้อมูลชนิดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำมาทำงานวิจัยและทดลอง และในวิทยานิพนธ์ฉบับนี้ยังได้นำเสนอวิธีการแปลงข้อมูลที่มีลักษณะเป็นโครงสร้างตารางหรือข้อมูลที่มีโครงสร้างความสัมพันธ์ไปเก็บเป็นข้อมูลคู่คีย์ผลลัพธ์บนระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

ในบทนี้จะอธิบายถึงขั้นตอนและวิธีการดำเนินงานในการออกแบบการทดลองทดสอบระบบฐานข้อมูลเชิงสัมพันธ์เปรียบเทียบกับระบบฐานข้อมูลแบบไม่สัมพันธ์ ทางผู้จัดทำได้ทำการศึกษาและรวบรวมทฤษฎีรวมถึงงานวิจัยที่เกี่ยวข้อง เพื่อนำมาใช้ในการวิเคราะห์และออกแบบการทดลอง และในบทนี้จะอธิบายถึงภาพรวมของการทดลองทั้งหมด รวมถึงอธิบายโครงสร้างแบบจำลองที่ใช้ในการทดลอง รวมถึงอธิบายถึงชุดข้อมูลที่นำมาใช้ในการทดลองและอุปกรณ์ที่ใช้การทดลอง

3.1 การศึกษาและวิเคราะห์ปัญหาของงานวิจัย

การศึกษาและรวบรวมข้อมูลเพื่อใช้สำหรับการทดลอง ทางผู้จัดทำได้ศึกษาข้อมูลทางด้านเทคโนโลยีที่นำมาใช้ในการจัดการและพัฒนาระบบ เช่น ภาษาหรือโปรแกรมที่ใช้ในการพัฒนารูปแบบหรือคุณลักษณะต่าง ๆ ของระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง รวมไปถึงการศึกษาค้นคว้าตามแหล่งข้อมูลต่าง ๆ เช่น ตำรา เอกสารวิทยานิพนธ์ทั้งในประเทศและนอกประเทศ หรือบทความตามเว็บไซต์ต่าง ๆ เป็นต้น และรวบรวมข้อมูลเกี่ยวกับข้อดีและข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ โดยสามารถสรุปได้ดังตารางที่ 3.1 และตารางที่ 3.2

ตารางที่ 3.1 สรุปข้อดีและข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์

ข้อดีของระบบฐานข้อมูลเชิงสัมพันธ์	ข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์
1. เก็บข้อมูลความสัมพันธ์ในรูปแบบตาราง ซึ่งประกอบด้วย แถวและสดมภ์ เป็นรูปแบบเก็บข้อมูลแบบ 2 มิติ ทำให้ทำความเข้าใจได้ง่ายไม่ซับซ้อน	1. ไม่สามารถเพิ่มหรือลดโครงสร้างระบบฐานข้อมูลได้โดยทันที ต้องมีขั้นตอนการเปลี่ยนแปลงโครงสร้างเสียก่อน
2. รองรับคุณสมบัติ ACID ซึ่งทำให้ได้ข้อมูลที่ถูกต้อง	2. การเพิ่มขีดความสามารถด้านประสิทธิภาพระบบฐานข้อมูลมีต้นทุนสูง
3. รูปแบบคำสั่งมีรูปแบบมาตรฐานและมีลักษณะโครงสร้างที่เข้าใจง่าย	3. สามารถเกิดการสูญเสียข้อมูลได้ง่าย เนื่องจากจัดเก็บข้อมูลในที่เดียวกัน และมีขั้นตอนการสำรองข้อมูลที่ซับซ้อนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 สรุปข้อดีและข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์(ต่อ)

ข้อดีของระบบฐานข้อมูลเชิงสัมพันธ์	ข้อเสียของระบบฐานข้อมูลเชิงสัมพันธ์
4. รูปแบบการเชื่อมความสัมพันธ์ของข้อมูลระหว่างตารางสามารถทำได้ง่าย เห็นภาพชัดเจน	4. เนื่องจากระบบฐานข้อมูลเชิงสัมพันธ์ไม่ได้ออกแบบเพื่อรองรับการประมวลผลแบบกระจายโดยตรง ดังนั้นการตั้งค่าระบบฐานข้อมูลให้ประมวลผลแบบกระจายจึงมีขั้นตอนที่ซับซ้อนมาก
5. เนื่องจากเป็นระบบฐานข้อมูลที่ได้รับความนิยมมากในปัจจุบัน จึงมีเครื่องมือหรือตัวช่วยในการจัดการระบบฐานข้อมูลรองรับมากมาย	5. เนื่องจากเป็นระบบฐานข้อมูลที่ได้รับความนิยมมาก จึงพบช่องโหว่ของระบบฐานข้อมูลอย่างหลากหลาย

ตารางที่ 3.2 สรุปข้อดีและข้อเสียของระบบฐานข้อมูลแบบไม่สัมพันธ์

ข้อดีของระบบฐานข้อมูลแบบไม่สัมพันธ์	ข้อเสียของระบบฐานข้อมูลแบบไม่สัมพันธ์
1. เป็นระบบฐานข้อมูลที่มีความยืดหยุ่นสูง	1. เนื่องจากเป็นระบบฐานข้อมูลที่เพิ่งถูกพัฒนาออกมาไม่นานนัก เพราะฉะนั้นจึงขาดความน่าเชื่อถือ
2. ออกแบบเพื่อรองรับการทำงานกับข้อมูลขนาดใหญ่ได้ดี	2. ขณะประมวลผลอาจจะใช้ทรัพยากรของเครื่องเยอะในตอนแรก
3. ขั้นตอนการขยายพื้นที่การจัดเก็บข้อมูลสามารถทำได้ง่ายมาก	3. ระบบฐานข้อมูลแบบไม่สัมพันธ์บางชนิดหรือบางเวอร์ชันอาจจะไม่รองรับคุณสมบัติของ ACID เหมือนระบบฐานข้อมูลเชิงสัมพันธ์ ทำให้ข้อมูลที่ได้อาจไม่ถูกต้อง 100 เปอร์เซ็นต์
4. การตั้งค่าระบบฐานข้อมูลให้ประมวลผลแบบกระจายมีขั้นตอนที่ไม่ซับซ้อน	4. ระบบฐานข้อมูลแบบไม่สัมพันธ์แต่ละแบบมีรูปแบบคำสั่งที่เฉพาะและแตกต่างกันออกไป ไม่มีมาตรฐานที่แน่นอน
5. รองรับการสำรองข้อมูล โดยมีขั้นตอนที่ไม่ซับซ้อน	5. เนื่องจากระบบฐานข้อมูลแบบไม่สัมพันธ์ยังไม่เป็นที่แพร่หลายกับนักพัฒนาระบบทั่วไป ทำให้เครื่องมือที่ช่วยในการจัดการระบบฐานข้อมูลจึงมีให้เลือกใช้ไม่มากนัก

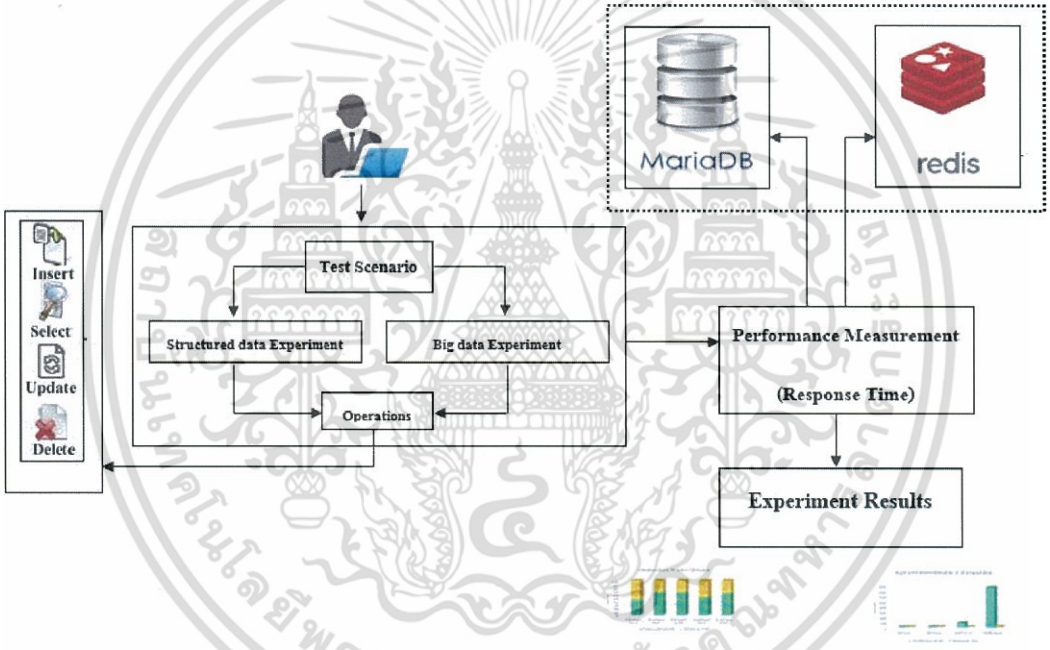
นอกจากนี้ผู้จัดทำยังได้ศึกษาภาษาพีเอชพี (PHP) ที่ใช้ในการออกแบบการทดลอง รวมถึงศึกษาการนำไลบรารี่ Predis มาใช้ในการทดลองเพื่อเชื่อมต่อระบบจัดการฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์อ้างอิง เรดิส (Redis) กับโปรแกรมภาษาพีเอชพีและได้ศึกษาองค์ประกอบรวมถึงรูปแบบการเก็บข้อมูลชนิดต่างของระบบจัดการฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงเรดิส โดยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่ใช้ได้แก่ มาเรียดีบี (MariaDB) และระบบจัดการฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่เลือกใช้ได้แก่ เรดิส

3.2 การวิเคราะห์และการออกแบบกรอบแนวคิด

การวิเคราะห์และออกแบบระบบภาพรวมของการทดสอบความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ แสดงให้เห็นดังรูปที่ 3.1



รูปที่ 3.1 กรอบแนวคิดของการวิจัย

จากภาพที่ 3.1 กรอบแนวคิดภาพรวมการวิจัย แสดงให้เห็นว่างานวิจัยฉบับนี้ทำการศึกษาออกแบบการทดลอง เพื่อเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) โดยใช้ฐานข้อมูลมาเรียดีบี และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง (Key Value Non- Relational Database) โดยใช้ฐานข้อมูลเรดิส โดยทางผู้จัดทำได้นำชุดข้อมูล 2 ชุดข้อมูล มาทำการทดลอง โดยชุดข้อมูลชุดแรกคือข้อมูลที่มีลักษณะเป็นโครงสร้าง (Structured Data) ซึ่งเป็นชุดข้อมูลที่มีความสัมพันธ์กันและชุดข้อมูลที่สองคือ ชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ (Big Data) โดยชุดข้อมูลชุดแรกเป็นชุดข้อมูลที่เหมาะกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลเชิงสัมพันธ์ โดยทางผู้จัดทำต้องการแสดงให้เห็นถึงประสิทธิภาพของระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เมื่อต้องประมวลผลข้อมูลในลักษณะที่มีโครงสร้าง และชุดข้อมูลชุดที่ 2 เป็นชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ ซึ่งข้อมูลชุดนี้เป็นชุดข้อมูลที่มีปริมาณมหาศาลและเป็นข้อมูลที่มีลักษณะของบิ๊กดาต้า (Big Data) โดยทางผู้จัดทำต้องการแสดงให้เห็นถึงประสิทธิภาพของฐานข้อมูลแต่ละชนิด เมื่อนำมาประมวลผลข้อมูลที่มีลักษณะของบิ๊กดาต้า เพราะฉะนั้นจึงออกแบบการทดลองออกเป็น 2 การทดลอง ได้แก่ การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยแต่ละการทดลองเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผล ประกอบด้วย 4 ตัวดำเนินการ ได้แก่ ความเร็วในการอ่าน (Select Operation) ความเร็วในการเขียน (Insert Operation) ความเร็วในการแก้ไข (Update Operation) และความเร็วในการลบ (Delete Operation) โดยผ่านชุดข้อมูลโครงสร้างเดียวกัน โดยแต่ละการทดลองของแต่ละตัวดำเนินการจะทำการวัดระยะเวลาการประมวลผลทั้งหมด 10 ครั้ง เพื่อหาค่าเฉลี่ยของเวลาในการประมวลผลของแต่ละการทดลอง ซึ่งระบบฐานข้อมูลเชิงสัมพันธ์ที่นำมาทำการทดลอง คือ มาเรียดีบี เวอร์ชัน 10.1.13-MariaDB และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่นำมาทำการทดลองคือ ฐานข้อมูล เรดิส เวอร์ชัน 3.2.100. โดยการทดลองงานวิจัยฉบับนี้ทำการทดลองอยู่บนระบบปฏิบัติการวินโดว์ (Windows) และทำการทดลองประมวลผลแบบเครื่องเดียว

3.3 ชุดข้อมูลที่ใช้ในงานวิจัย

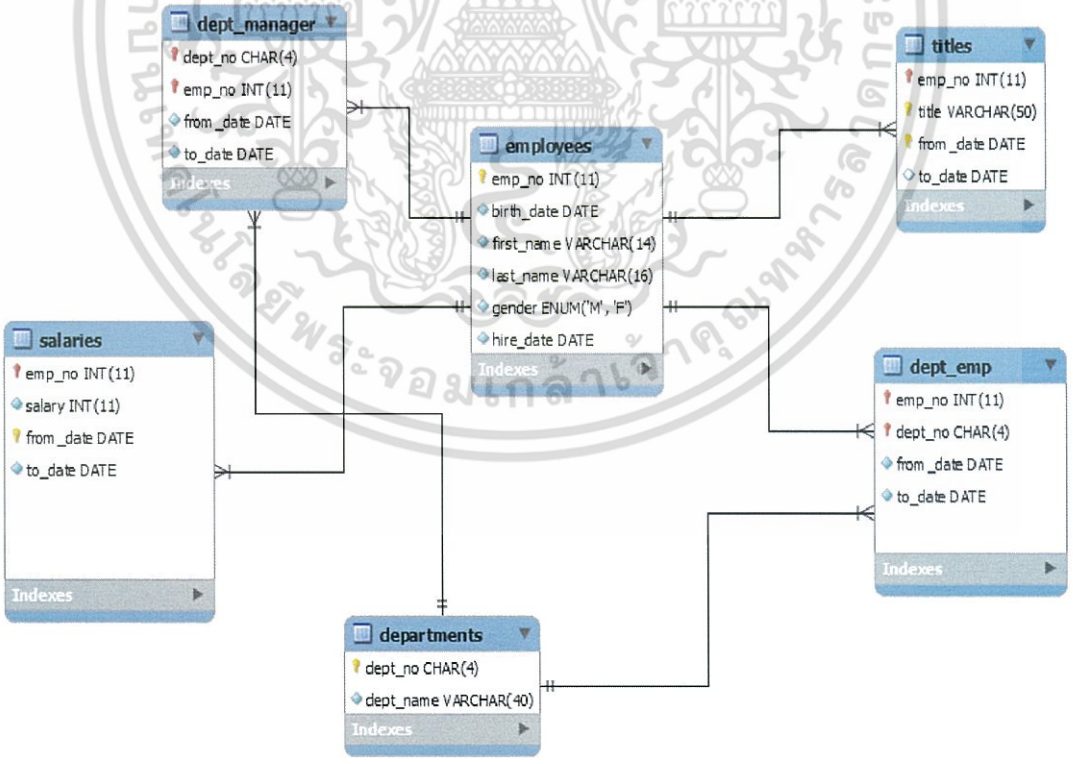
ในงานวิจัยฉบับนี้ได้ทำการทดลองบนชุดข้อมูล 2 ชุดข้อมูล ได้แก่ ชุดข้อมูลที่มีลักษณะเป็นโครงสร้างและชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

3.3.1 ชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

ชุดข้อมูลที่มีลักษณะเป็นโครงสร้างเป็นชุดข้อมูลแสดงข้อมูลของพนักงานโดยประกอบด้วยข้อมูล 6 ตารางหรือออบเจ็ค ซึ่งประกอบด้วยตาราง employees dept_manager departments salaries titles และ dept_emp โดยตาราง employees ประกอบไปด้วยข้อมูลจำนวน 300,024 ระเบียน และตาราง employees ประกอบด้วย 6 แอททริบิวต์ ได้แก่ emp_no birth_date first_name last_name gender และ hire_date ซึ่งตารางหรือออบเจ็ค employees นี้มี emp_no เป็นคีย์หลัก สำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิงสัมพันธ์ และตาราง dept_manager ประกอบไปด้วยข้อมูลจำนวน 24 ระเบียน ประกอบด้วย 4 แอททริบิวต์ ได้แก่ emp_no dept_no from_date และ to_date ซึ่งตารางหรือออบเจ็ค dept_manager นี้มี emp_no และ dept_no เป็นคีย์หลักและคีย์นอก สำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

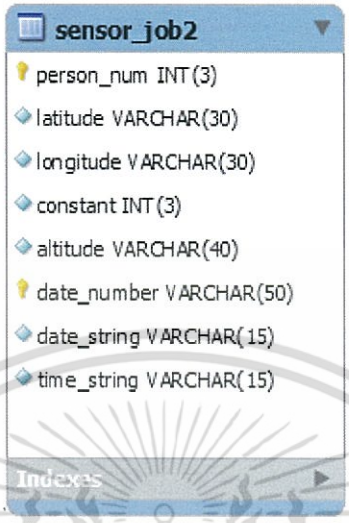
สัมพันธ์ ส่วนตาราง departments ประกอบไปด้วยข้อมูลจำนวน 9 ระเบียน ประกอบด้วย 2 แอททริบิวต์ ได้แก่ dept_no และ dept_name โดยมี dept_no เป็นคีย์หลัก สำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิงสัมพันธ์ ส่วนตาราง salaries ประกอบไปด้วยข้อมูลจำนวน 2,844,047 ระเบียน ประกอบด้วย 4 แอททริบิวต์ ได้แก่ emp_no salary from_date และ to_date โดยมี emp_no และ from_date เป็นคีย์หลัก และมี emp_no เป็นคีย์นอก สำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิงสัมพันธ์ ตาราง titles ประกอบด้วยข้อมูลจำนวน 443,308 ระเบียน ประกอบด้วย 4 แอททริบิวต์ ได้แก่ emp_no title from_date และ to_date โดยมี emp_no title และ from_date ถูกกำหนดเป็นคีย์หลักและมี emp_no ถูกกำหนดเป็นคีย์นอก สำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิงสัมพันธ์ และตาราง dept_emp ประกอบด้วยข้อมูลจำนวน 331,603 ระเบียน ซึ่งประกอบด้วย 4 แอททริบิวต์ ได้แก่ emp_no dept_no from_date และ to_date โดยมี emp_no และ dept_no ถูกกำหนดเป็นคีย์หลักและคีย์นอก สำหรับการกำหนดโครงสร้างบนระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งจากรูปภาพแบบจำลองโครงสร้างฐานข้อมูลแสดงให้เห็นว่าข้อมูลมีความสัมพันธ์กันระหว่างข้อมูลดังรูปที่ 3.2 ซึ่งทางผู้จัดทำได้ทำการดาวน์โหลดชุดข้อมูลจำลองมาจาก test suite 1.0.6 [12]



รูปที่ 3.2 แบบจำลองชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่



รูปที่ 3.3 แบบจำลองชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

ชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ เป็นข้อมูลแสดงพิกัดตำแหน่งที่รับจากเซ็นเซอร์จีพีเอส (GPS) ของผู้ใช้งาน (User) และข้อมูลชุดนี้ยังแสดงข้อมูลที่เป็นเวลาที่รับจากเซ็นเซอร์จีพีเอส ซึ่งประกอบไปด้วย 8 แอททริบิวต์ ได้แก่ person_num latitude longitude constant altitude date_number date_string และ time_string โดย person_num และ date_number ถูกกำหนดให้เป็นคีย์หลักสำหรับการกำหนดโครงสร้างบนฐานข้อมูลเชิงสัมพันธ์ ซึ่งชุดข้อมูลชุดนี้เป็นชุดข้อมูลที่มีลักษณะของบิ๊กดาต้า โดยทางผู้จัดทำได้ทำการดาวน์โหลดชุดข้อมูลจำลองชุดนี้มาจาก Microsoft Research Asia Geolife project [27] ซึ่งข้อมูลชุดนี้ประกอบด้วยข้อมูลจำนวน 1,844,244 ระเบียน ซึ่งแสดงแบบจำลองข้อมูลดังรูปภาพที่ 3.3

3.4 ตัวดำเนินการ

ในวิทยานิพนธ์ฉบับนี้ได้แบ่งการทดลองออกเป็น 4 ตัวดำเนินการ ซึ่งประกอบด้วย การเขียน การลบ การแก้ไขและการอ่าน โดยการทดลองในแต่ละตัวดำเนินการได้ทำการทดสอบกับชุดข้อมูล 2 ชุดข้อมูล ได้แก่ ชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

3.4.1 การเขียน

การทดลองเขียนข้อมูลจะทำการทดสอบกับชุดข้อมูล 2 ชุดข้อมูล คือ การเขียนสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการเขียนสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.1 การเขียนสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

การทดสอบความสามารถในการเขียนของระบบฐานข้อมูลทั้ง 2 ชนิด สำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง ทำการเขียนข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียนและ 100,000 ระเบียน ลงบนระบบฐานข้อมูลทั้ง 2 ชนิด โดยทางผู้จัดทำได้เลือกใช้ตารางหรือแอนติตี้ Salaries ในการทดลองนี้ ซึ่งในการทดลองการเขียนข้อมูลได้แบ่งรูปแบบการเขียนข้อมูลออกเป็น 2 รูปแบบ คือ การเขียนข้อมูลลงบนระบบฐานข้อมูลแบบที่ละคำสั่ง (Single Write) และการเขียนข้อมูลแบบรวมคำสั่ง (Multiple Write) โดยในระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่ชื่อว่า เรดิส มีฟังก์ชันพิเศษที่เรียกว่า Pipeline Feature ซึ่งฟังก์ชันพิเศษที่ทำให้สามารถรวมคำสั่งหลาย ๆ คำสั่งที่ต่อเนื่องกัน แล้วส่งให้ฐานข้อมูลเรดิส ทำงานเพียงครั้งเดียว โดยไม่ต้องรอให้เซิร์ฟเวอร์ตอบกลับการร้องขอของแต่ละคำสั่งก่อน ซึ่งเป็นฟังก์ชันการเขียนข้อมูลแบบรวมคำสั่งของฐานข้อมูลเรดิส ซึ่งเป็นฟังก์ชันพิเศษที่จำเป็นมากสำหรับการใช้งานฐานข้อมูลเรดิส ดังนั้นเพื่อวัดประสิทธิภาพในการเขียน คำสั่งที่ใช้ในการเขียนข้อมูลของระบบฐานข้อมูลทั้ง 2 ชนิดจะต้องมีความหมายและให้ผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.4.1.2 การเขียนสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

การทดสอบความสามารถในการเขียนของระบบฐานข้อมูลทั้ง 2 ชนิด สำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยทำการเขียนข้อมูลลงบนฐานข้อมูลทั้ง 2 ชนิด จำนวน 1,844,244 ระเบียน ซึ่งการทดลองนี้ใช้การเขียนข้อมูลแบบรวมคำสั่ง นอกจากนี้การทดลองการเขียนสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่นี้ ได้ทำการเขียนข้อมูล 3 ขนาด ลงบนฐานข้อมูลทั้ง 2 ชนิด โดยแบ่งเป็น ข้อมูลขนาด 500,000 ระเบียน 1,000,000 ระเบียนและ 1,500,000 ระเบียนตามลำดับ เพื่อสังเกตถึงประสิทธิภาพลด้านการเขียนของฐานข้อมูลทั้ง 2 ชนิด เมื่อปริมาณของข้อมูลเพิ่มขึ้น ดังนั้นเพื่อวัดประสิทธิภาพในการเขียน คำสั่งที่ใช้ในการเขียนข้อมูลของระบบฐานข้อมูลทั้ง 2 ชนิดจะต้องมีความหมายและให้ผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.4.2 การลบ

การทดลองลบข้อมูลจะทำการทดสอบกับชุดข้อมูล 2 ชุดข้อมูล คือ การลบสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการลบสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

3.4.2.1 การลบสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

การทดสอบความสามารถในการลบข้อมูลที่มีลักษณะเป็นโครงสร้าง โดยการทดลองการลบข้อมูลได้แบ่งการทดลองออกเป็น 2 ลักษณะ โดยการลบข้อมูลในลักษณะที่ 1 จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนึงถึงปริมาณข้อมูลเป็นสำคัญ การลบข้อมูลลักษณะที่ 1 จะทำการลบข้อมูลจากระบบฐานข้อมูลทั้งสองชนิด โดยพิจารณาลบข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน โดยทางผู้จัดทำได้เลือกใช้ตารางหรือแอนติตี้ Salaries ในการทำการทดลองลบข้อมูลลักษณะที่ 1 และการทดลองลักษณะที่ 2 คือการลบข้อมูลโดยพิจารณาอ้างอิงถึงข้อจำกัดหรือเงื่อนไขเฉพาะเป็นสำคัญ หรือเรียกว่าเป็นการลบข้อมูลแบบมีเงื่อนไข ดังนั้นเพื่อวัดประสิทธิภาพในการลบข้อมูล คำสั่งที่ใช้ในการทดลองลบข้อมูลของทั้ง 2 ระบบฐานข้อมูลจะต้องให้ความหมายและผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

เงื่อนไขที่ใช้ในการทดลองลบข้อมูลลักษณะที่ 2 มีรายละเอียดดังนี้

เงื่อนไขที่ 1 : ลบข้อมูลจากตาราง salaries ที่มี emp_no = '10004'

เงื่อนไขที่ 2 : ลบข้อมูลจากตาราง departments ที่มี dept_no = 'd009'

3.4.2.2 การลบสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

การทดสอบความสามารถในการลบของระบบฐานข้อมูลทั้ง 2 ชนิด สำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยทำการลบข้อมูลลงบนฐานข้อมูลทั้ง 2 ชนิด จำนวน 1,844,244 ระเบียน นอกจากนี้การทดลองการลบสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่นี้ได้ทำการลบข้อมูล 3 ขนาด จากฐานข้อมูลทั้ง 2 ชนิด โดยแบ่งเป็น ข้อมูลขนาด 500,000 ระเบียน 1,000,000 ระเบียน และ1,500,000 ระเบียน ตามลำดับ เพื่อสังเกตถึงประสิทธิภาพลด้านการลบของฐานข้อมูลทั้ง 2 ชนิด เมื่อปริมาณของข้อมูลเพิ่มขึ้น ดังนั้นเพื่อวัดประสิทธิภาพในการลบ คำสั่งที่ใช้ในการลบข้อมูลของระบบฐานข้อมูลทั้ง 2 ชนิดจะต้องมีความหมายและให้ผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.4.3 การแก้ไข

การทดลองแก้ไขข้อมูลจะทำการทดสอบกับชุดข้อมูล 2 ชุดข้อมูล คือ การแก้ไขสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการแก้ไขสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

3.4.3.1 การแก้ไขสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

การทดสอบความสามารถในการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้าง โดยการทดลองการแก้ไขข้อมูลได้แบ่งการทดลองออกเป็น 2 ลักษณะ โดยการแก้ไขข้อมูลในลักษณะที่ 1 จะคำนึงถึงปริมาณข้อมูลเป็นสำคัญ การแก้ไขข้อมูลลักษณะที่ 1 จะทำการแก้ไขข้อมูลจากระบบฐานข้อมูลทั้งสองชนิด โดยพิจารณาแก้ไขข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน โดยทางผู้จัดทำได้เลือกใช้ตารางหรือแอนติตี้ Salaries ในการทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองแก้ไขข้อมูลลักษณะที่ 1 และการทดลองลักษณะที่ 2 คือการแก้ไขข้อมูลโดยพิจารณาอ้างอิงถึงข้อจำกัดหรือเงื่อนไขเฉพาะเป็นสำคัญ หรือเรียกว่าเป็นการแก้ไขข้อมูลแบบมีเงื่อนไข ดังนั้นเพื่อวัดประสิทธิภาพในการแก้ไขข้อมูล คำสั่งที่ใช้ในการทดลองแก้ไขข้อมูลของทั้ง 2 ระบบฐานข้อมูลจะต้องให้ความหมายและผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

เงื่อนไขที่ใช้ในการทดลองแก้ไขข้อมูลลักษณะที่ 2 มีรายละเอียดดังนี้

เงื่อนไขที่ 1 : แก้ไขข้อมูลจากตาราง departments ที่มี dept_no = 'd001'

เงื่อนไขที่ 2 : แก้ไขข้อมูลจากตาราง titles ซึ่งทำการแก้ไขข้อมูล title และ to_date เมื่อมี emp_no = 10277 และมี from_date = 1985-06-16

3.4.3.2 การแก้ไขสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

การทดสอบความสามารถในการแก้ไขของระบบฐานข้อมูลทั้ง 2 ชนิด สำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยทำการแก้ไขข้อมูลลงบนฐานข้อมูลทั้ง 2 ชนิด จำนวน 1,844,244 ระเบียน นอกจากนี้การทดลองการแก้ไขสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่นี้ ได้ทำการแก้ไขข้อมูล 3 ขนาด จากฐานข้อมูลทั้ง 2 ชนิด โดยแบ่งเป็น ข้อมูลขนาด 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ เพื่อสังเกตถึงประสิทธิภาพลด้านการแก้ไขของฐานข้อมูลทั้ง 2 ชนิด เมื่อปริมาณของข้อมูลเพิ่มขึ้น ดังนั้นเพื่อวัดประสิทธิภาพในการแก้ไข คำสั่งที่ใช้ในการแก้ไขข้อมูลของระบบฐานข้อมูลทั้ง 2 ชนิดจะต้องมีความหมายและให้ผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.4.4 การอ่าน

การทดลองอ่านข้อมูลจะทำการทดสอบกับชุดข้อมูล 2 ชุดข้อมูล คือ การอ่านสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการอ่านสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

3.4.4.1 การอ่านสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

การทดลองอ่านข้อมูลที่มีลักษณะเป็นโครงสร้าง ในงานวิจัยฉบับนี้ได้แบ่งการทดลองการอ่านข้อมูลออกเป็น 2 ลักษณะ โดยการอ่านในลักษณะที่ 1 จะคำนึงถึงปริมาณของข้อมูลเป็นสำคัญ โดยลักษณะที่ 1 จะทำการอ่านข้อมูลจากระบบฐานข้อมูลทั้ง 2 ชนิด โดยพิจารณาอ่านข้อมูลที่มีจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน โดยทางผู้จัดทำได้เลือกใช้ตารางหรือแอนติตี้ Salaries ในการทดลองการอ่านข้อมูลลักษณะที่ 1 และการทดลองการอ่านข้อมูลลักษณะที่ 2 คือ การอ่านข้อมูลแบบคำนึงถึงข้อจำกัดและเงื่อนไขเฉพาะ หรือสามารถเรียกว่าเป็นการอ่านแบบมีเงื่อนไข ซึ่งจะแบ่งรูปแบบการอ่านข้อมูลลักษณะที่ 2 ออกเป็น 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ คือการอ่านข้อมูลแบบอย่างง่ายและการอ่านข้อมูลแบบซับซ้อน โดยการอ่านข้อมูลแบบอย่างง่าย คือการอ่านข้อมูลแบบมีเงื่อนไขจากตารางข้อมูล 1 ตาราง ส่วนการอ่านข้อมูลแบบซับซ้อน คือการอ่านข้อมูลแบบมีเงื่อนไขจากตารางข้อมูลมากกว่า 1 ตารางขึ้นไป โดยตัวดำเนินการจอย (Join Operation) และตัวดำเนินการซับคิวรี (Subquery Operation) ถูกใช้ในการอ่านข้อมูลแบบซับซ้อนในฐานข้อมูลมาเรียดีบี ซึ่งเป็นการอ่านข้อมูลที่มีความสัมพันธ์กัน

โดยการทดลองในการอ่านข้อมูลในลักษณะที่ 2 นี้จะประกอบด้วยเงื่อนไขการอ่านข้อมูลตามลำดับ ดังนี้

1. อ่านข้อมูลส่วนตัวของพนักงาน ที่มีเพศเป็นผู้ชาย (อย่างง่าย)
2. อ่านข้อมูลจากตาราง dept_emp ที่มีรหัสพนักงาน = 10021 (อย่างง่าย)
3. อ่านข้อมูลจากตาราง dept_emp และตาราง departments เพื่อหาว่าพนักงานแต่ละคนอยู่ในแผนกไหนบ้าง และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)
4. อ่านข้อมูลจากตาราง departments และ dept_manager เพื่อหาว่า พนักงานที่มีตำแหน่ง ระดับพนักงาน อยู่ในแผนกไหนบ้าง และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)
5. อ่านข้อมูลส่วนตัวของพนักงานและเป็นพนักงานที่เป็นระดับผู้จัดการ และหาว่าพนักงานคนนั้นอยู่ในแผนกไหน ซึ่งทำการอ่านข้อมูลจาก 3 ตาราง คือ ตาราง employees ตาราง dept_manager และตาราง departments และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)
6. อ่านข้อมูลส่วนตัวของพนักงาน ซึ่งมีตำแหน่งเป็นระดับผู้จัดการ และมีรหัส dept_no = “d001” ซึ่งทำการอ่านข้อมูลจากตาราง employees และตาราง dept_manager โดยใช้รูปแบบการอ่านข้อมูลแบบซับคิวรี่บนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)
7. อ่านข้อมูลส่วนตัวของพนักงาน ซึ่งมีเงินเดือนน้อยกว่าหรือเท่ากับ 50,000 บาท ซึ่งทำการอ่านข้อมูลจากตาราง employees และตาราง salaries โดยใช้รูปแบบการอ่านข้อมูลแบบซับคิวรี่บนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)

ดังนั้นเพื่อวัดประสิทธิภาพในการอ่านข้อมูล คำสั่งที่ใช้ในการทดลองอ่านข้อมูลของทั้ง 2 ระบบฐานข้อมูลจะต้องให้ความหมายและผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.4.4.2 การอ่านสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

การทดสอบความสามารถในการอ่านของระบบฐานข้อมูลทั้ง 2 ชนิด สำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยทำการอ่านข้อมูลลงบนฐานข้อมูลทั้ง 2 ชนิด จำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1,844,244 ระเบียน นอกจากนี้การทดลองการอ่านสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่นี้ ได้ทำการอ่านข้อมูล 3 ขนาด จากฐานข้อมูลทั้ง 2 ชนิด โดยแบ่งเป็น ข้อมูลขนาด 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ เพื่อสังเกตถึงประสิทธิภาพด้านการอ่านของฐานข้อมูลทั้ง 2 ชนิด เมื่อปริมาณของข้อมูลเพิ่มขึ้น ดังนั้นเพื่อวัดประสิทธิภาพในการอ่าน คำสั่งที่ใช้ในการอ่านข้อมูลของระบบฐานข้อมูลทั้ง 2 ชนิดจะต้องมีความหมายและให้ผลลัพธ์ที่เหมือนกัน โดยจะแสดงผลการเปรียบเทียบประสิทธิภาพทางเวลาของระบบฐานข้อมูลทั้ง 2 ชนิดในหัวข้อของการทดลองและผลการทดลอง

3.5 เครื่องมือในการทดลอง

3.5.1 ด้านฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์หน่วยประมวลผล Intel Core i5-2450M 2.5 GHz
- หน่วยความจำหลัก 8.00 GB
- ฮาร์ดดิสก์มีความจุ 1 TB

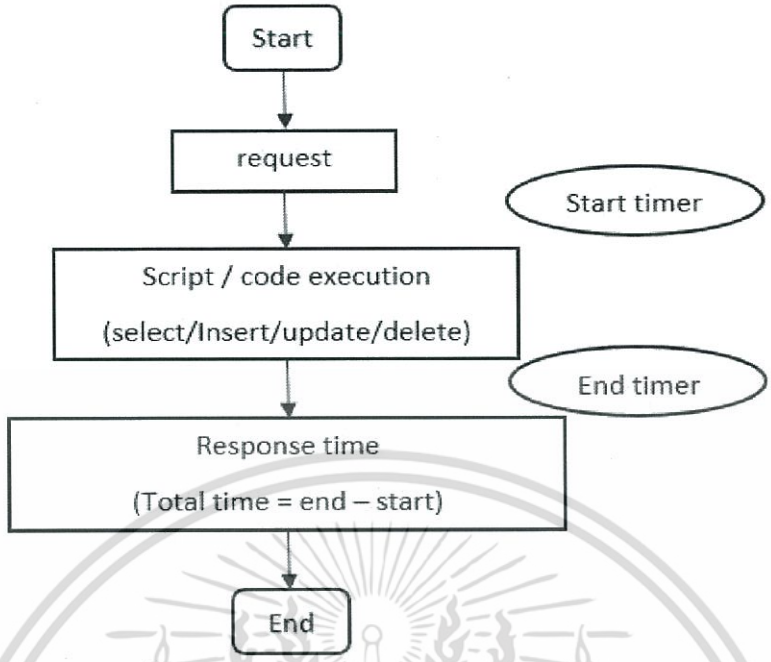
3.5.2 ด้านซอฟต์แวร์ (Software)

- ระบบปฏิบัติการวินโดว์ (Windows) 10 แบบ 64 บิต
- ฐานข้อมูลเชิงสัมพันธ์มาเรียดีบี เวอร์ชัน 10.1.13-MariaDB
- ฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงเรดิส เวอร์ชัน 3.2.100.
- Sublime Text 3114
- XAMPP Web Server Control Panel v.3.2.2
- ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรม คือ ภาษาพีเอชพี

3.6 สถิติที่นำมาใช้ในงานวิจัย

กระบวนการวัดค่าเวลาการประมวลผลของระบบฐานข้อมูลแต่ละแบบ เริ่มจากการส่งคำร้องขอไปยังฐานข้อมูล และจากนั้นประมวลผลตัวดำเนิดการแต่ละตัว ที่มีการออกแบบการทดลองไว้ คือ การเขียน การแก้ไข การลบ การอ่าน บันทึกเวลาหลังการประมวลผลเสร็จจะได้เวลาที่ฐานข้อมูลใช้สำหรับการประมวลผล โดยมีกระบวนบันทึกเวลาดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การวัดระยะเวลาในการประมวลผลข้อมูล

การคำนวณเวลาที่ใช้วัดความเร็วในการประมวลผลของฐานข้อมูลแต่ละแบบมีการคำนวณดังสมการที่ 1

$$\text{Total time} = \text{end} - \text{start} \quad (1)$$

เมื่อ Total time คือเวลาทั้งหมดที่ใช้ในการประมวลผล

End คือ เวลาเมื่อมีการประมวลผลเสร็จสิ้น

Start คือ เวลาเริ่มต้นในการประมวลผลข้อมูล

ซึ่งการทดสอบในแต่ละตัวดำเนินการจะทำการทดลอง 10 ครั้ง และหาค่าระยะเวลาเฉลี่ย โดยการทดลองในงานวิจัยชุดนี้ใช้ ฟังก์ชันพีเอชพีไมโครไทม์ (PHP Function Microtime) เพื่อใช้ในการวัดระยะเวลาในการประมวลผล

3.7 การแปลงข้อมูลที่มีลักษณะเป็นตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์

ฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เป็นฐานข้อมูลชนิดหนึ่งที่มีลักษณะไม่เป็นโครงสร้างและไม่มีความสัมพันธ์ ซึ่งแตกต่างจากฐานข้อมูลเชิงสัมพันธ์โดยสิ้นเชิง ซึ่งฐานข้อมูลดังกล่าวยังถือว่าเป็นฐานข้อมูลที่มีรูปแบบการเก็บข้อมูลที่หลากหลายและถือว่าเป็นฐานข้อมูลชนิดหนึ่งที่มีความยืดหยุ่นสูง ซึ่งฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงนั้นเก็บข้อมูลอยู่ในรูปแบบคู่คีย์ผลลัพธ์ โดยรูปแบบคู่คีย์ผลลัพธ์สามารถแบ่งรูปแบบการจัดเก็บข้อมูลได้หลากหลายรูปแบบ ประกอบด้วย Sets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lists Sorted Sets Strings และ Hashes เป็นต้น ซึ่งขึ้นอยู่กับผู้ใช้งาน ว่าจะเลือกใช้รูปแบบการเก็บข้อมูลแบบไหนหรือจะออกแบบรูปแบบการเก็บข้อมูลเป็นลักษณะใด ดังนั้นการแปลงข้อมูลที่มีลักษณะเป็นตารางไปเป็นข้อมูลคู่คีย์ผลลัพธ์บนฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง จะต้องมีการออกแบบรูปแบบการเก็บข้อมูลลักษณะคู่คีย์ผลลัพธ์เสียก่อน ซึ่งในงานวิจัยของวิทยานิพนธ์ฉบับนี้ทางผู้จัดทำได้ทำการคิดค้นและออกแบบวิธีการแปลงข้อมูลที่มีลักษณะเป็นตารางหรือเป็นข้อมูลที่มีลักษณะของโครงสร้างไปเป็นข้อมูลที่เป็นลักษณะคู่คีย์ผลลัพธ์ โดยทางผู้จัดทำได้แบ่งรูปแบบการแปลงข้อมูลออกเป็น 2 รูปแบบ ได้แก่ 1. รูปแบบที่พิจารณาจากโครงสร้างตาราง 2. รูปแบบที่พิจารณาจากข้อมูลในตาราง โดยวิธีการออกแบบรูปแบบการเก็บข้อมูลมีรายละเอียดดังต่อไปนี้

3.7.1 รูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากโครงสร้างตาราง

รูปแบบจากโครงสร้างตาราง เป็นรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์เพื่อแสดงรายละเอียดโครงสร้างของตารางข้อมูล โดยข้อมูลโครงสร้างตารางที่สำคัญที่ต้องพิจารณา อาทิเช่น ตารางนี้มีคีย์หลักเป็นแอททริบิวต์ไหน และคีย์หลักในตารางนี้เป็นคีย์นอกในตารางอื่นหรือไม่ เป็นต้น ซึ่งการแปลงข้อมูลที่พิจารณาจากโครงสร้างตาราง จะทำการพิจารณาถึง ชื่อตาราง ชื่อแอททริบิวต์ของตาราง คีย์หลักของตาราง คีย์นอกของตาราง และพิจารณาความสัมพันธ์ระหว่างตาราง เพื่อแปลงข้อมูลหรือรายละเอียดดังกล่าวในอยู่ในรูปแบบของข้อมูลคู่คีย์ผลลัพธ์บนฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยการแปลงข้อมูลดังกล่าวมีวิธีการดังต่อไปนี้

1. พิจารณาถึงชื่อของตารางข้อมูลและชื่อของแอททริบิวต์แต่ละตัว
2. สร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets เพื่อเก็บข้อมูลรายละเอียดโครงสร้างตาราง
3. ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยตั้งชื่อจากการนำชื่อตารางที่กำลังพิจารณา มาใส่เป็นชื่อคีย์ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ดังกล่าว
4. ใส่ค่าผลลัพธ์ ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยนำชื่อแอททริบิวต์ทุกตัวของตารางข้อมูล บรรจุลงไปเป็นค่าผลลัพธ์สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ดังกล่าว
5. พิจารณาถึงชื่อแอททริบิวต์ที่เป็นคีย์หลักของตารางข้อมูล
6. สร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets เพื่อเก็บค่าของคีย์หลักจากตารางข้อมูล
7. ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 6 โดยนำชื่อตารางดังกล่าว มาระบุเป็นชื่อคีย์ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets และระบุตามด้วยคำว่า ": Primary Key"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ใส่ค่าผลลัพธ์ ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 6 โดยใส่ชื่อแอททริบิวต์ที่เป็นคีย์หลักของตารางดังกล่าวลงไปในช่องข้อมูลผลลัพธ์ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 6

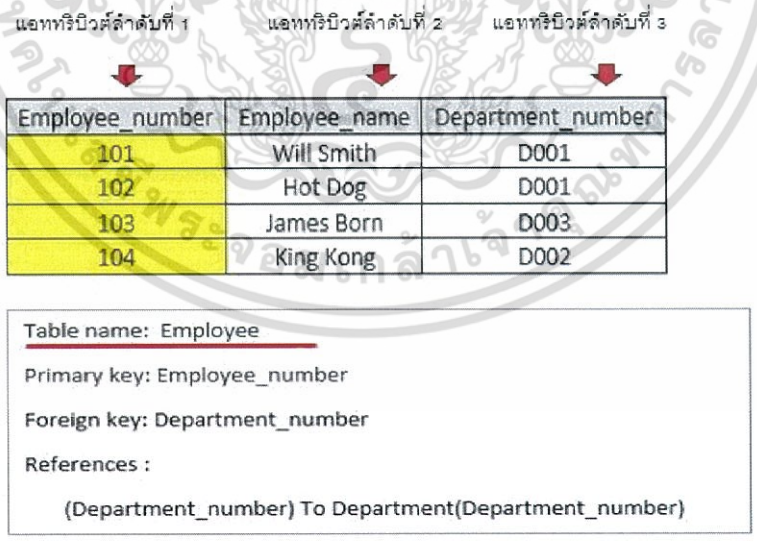
9. พิจารณาความสัมพันธ์ระหว่างตาราง โดยทำการหาค่าคีย์นอกในตาราง หากพบคีย์นอก ให้สร้างชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ขึ้นมาเพื่อเก็บรายละเอียดความสัมพันธ์ระหว่างตาราง

10. ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อตารางที่มีความสัมพันธ์ร่วม มาตั้งเป็นชื่อคีย์และนำชื่อแอททริบิวต์ที่ถูกอ้างอิงความสัมพันธ์มาจากตารางที่มีความสัมพันธ์ร่วม มาตั้งเป็นชื่อคีย์ด้วยเช่นกัน โดยขั้นด้วยเครื่องหมาย ":" ซึ่งมีลักษณะดังนี้ "[ชื่อตารางที่มีความสัมพันธ์ร่วม]:[ชื่อแอททริบิวต์ที่ถูกอ้างอิงความสัมพันธ์]"

11. ตั้งชื่อค่าของข้อมูล Field สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อตารางที่กำลังพิจารณาอยู่นี้มาตั้งเป็นค่าของข้อมูล Field ดังกล่าว

12. ระบุค่าข้อมูลผลลัพธ์สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อแอททริบิวต์ที่เป็นคีย์นอกของตารางดังกล่าวมาระบุเป็นค่าผลลัพธ์

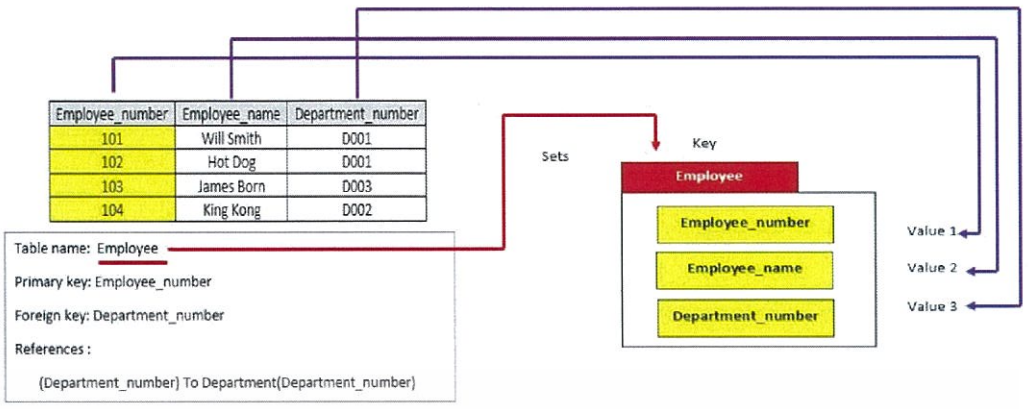
เพื่อสร้างความเข้าใจสำหรับวิธีการแปลงข้อมูลในรูปแบบที่ 1 มากยิ่งขึ้น ทางผู้จัดทำจึงได้นำเสนอวิธีการแปลงข้อมูลในรูปแบบที่ 1 นี้ผ่านตัวอย่างดังต่อไปนี้



รูปที่ 3.5 การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 1

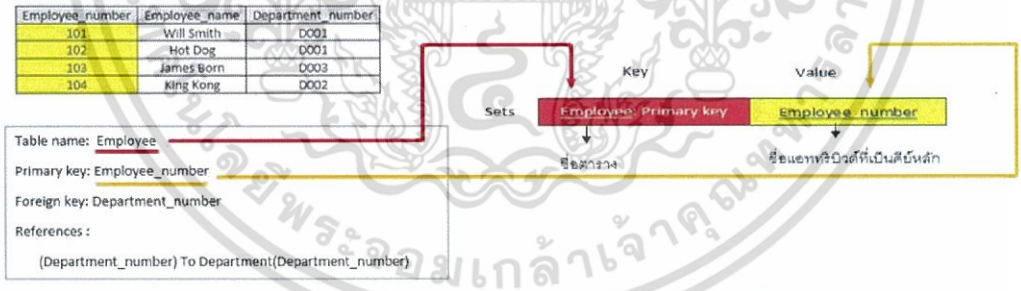
ในขั้นตอนที่ 1 พิจารณาถึงชื่อของตารางข้อมูลและชื่อของแอททริบิวต์แต่ละตัว ดังแสดงในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 2 3 และ 4

ในขั้นตอนที่ 2 สร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets เพื่อเก็บข้อมูลรายละเอียดโครงสร้างตาราง และในขั้นตอนที่ 3 ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยตั้งชื่อจากการนำชื่อตารางที่กำลังพิจารณา มาใส่เป็นชื่อคีย์ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ดังกล่าว และในขั้นตอนที่ 4 ใส่ค่าผลลัพธ์ ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยนำชื่อแอททริบิวต์ทุกตัวของตารางข้อมูล บรรจุลงไปเป็นค่าผลลัพธ์สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ดังกล่าว ซึ่งแสดงให้เห็นดังรูปที่ 3.6

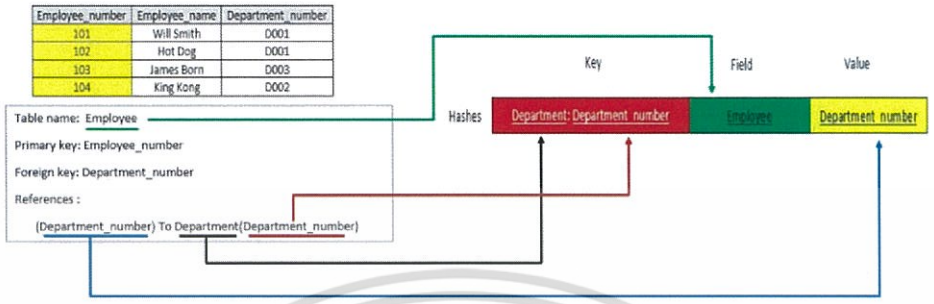


รูปที่ 3.7 การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 5 6 7 และ 8

ในขั้นตอนที่ 5 พิจารณาถึงชื่อแอททริบิวต์ที่เป็นคีย์หลักของตารางข้อมูล ต่อมาในขั้นตอนที่ 6 สร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets เพื่อเก็บค่าของคีย์หลักจากตารางข้อมูล จากนั้นในขั้นตอนที่ 7 ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 6 โดยนำชื่อตารางดังกล่าว มาระบุเป็นชื่อคีย์ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets และระบุตามด้วยคำว่า ": Primary Key" และในขั้นตอนที่ 8 ใส่ค่าผลลัพธ์ ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจาก

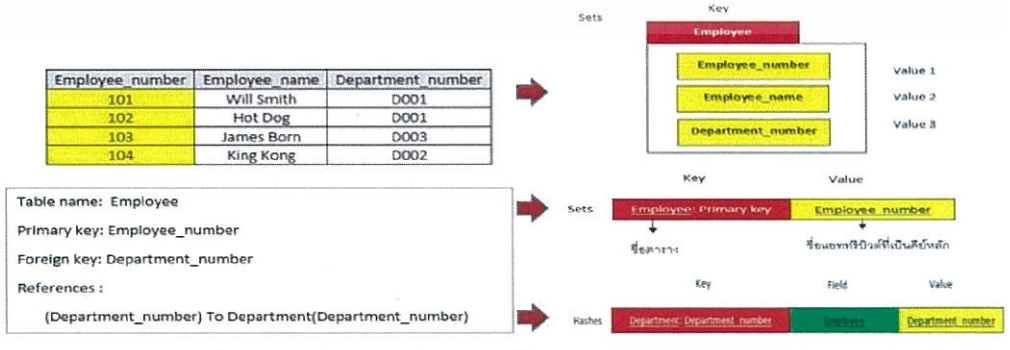
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 6 โดยใส่ชื่อแอททริบิวต์ที่เป็นคีย์หลักของตารางดังกล่าวลงไปในช่องข้อมูลผลลัพธ์ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Sets ที่สร้างขึ้นมาจากขั้นตอนที่ 6 ดังแสดงตัวอย่างในรูปที่ 3.7



รูปที่ 3.8 การแปลงข้อมูลโครงสร้างเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 9 10 11 และ 12

ในขั้นตอนที่ 9 พิจารณาความสัมพันธ์ระหว่างตาราง โดยทำการหาค่าคีย์นอกในตาราง หากพบคีย์นอก ให้สร้างชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ขึ้นมาเพื่อเก็บรายละเอียดความสัมพันธ์ระหว่างตาราง ต่อมาในขั้นตอนที่ 10 ตั้งชื่อคีย์ ของข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อตารางที่มีความสัมพันธ์ร่วม มาตั้งเป็นชื่อคีย์และนำชื่อแอททริบิวต์ที่ถูกอ้างอิงความสัมพันธ์มาจากตารางที่มีความสัมพันธ์ร่วม มาตั้งเป็นชื่อคีย์ด้วยเช่นกัน โดยขั้นด้วยเครื่องหมาย ":" ซึ่งมีลักษณะดังนี้ "[ชื่อตารางที่มีความสัมพันธ์ร่วม]:[ชื่อแอททริบิวต์ที่ถูกอ้างอิงความสัมพันธ์]" จากนั้นในขั้นตอนที่ 11 ตั้งชื่อค่าของข้อมูล Field สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อตารางที่กำลังพิจารณาอยู่นี้มาตั้งเป็นค่าของข้อมูล Field ดังกล่าว และในขั้นตอนที่ 12 ระบุค่าข้อมูลผลลัพธ์สำหรับชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 9 โดยนำชื่อแอททริบิวต์ที่เป็นคีย์นอกของตารางดังกล่าวมาระบุเป็นค่าผลลัพธ์ของชุดข้อมูล Hashes ดังแสดงในรูปที่ 3.8 โดยในรูปที่ 3.9 เป็นรูปภาพแสดงชุดข้อมูลคู่คีย์ผลลัพธ์ทั้งหมดที่พิจารณาจากข้อมูลโครงสร้างของตารางตัวอย่าง



รูปที่ 3.9 ชุดข้อมูลคู่คีย์ผลลัพธ์ที่พิจารณาจากโครงสร้างตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 รูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์จากข้อมูลตาราง

รูปแบบจากข้อมูลตาราง เป็นรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์เพื่อแสดงรายละเอียดจากตารางข้อมูล โดยจะทำการแปลงข้อมูลที่เก็บอยู่ในตาราง ไปเก็บข้อมูลในฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยใช้ประโยชน์จากความหลากหลายในการเก็บข้อมูลข้อมูลคู่คีย์ผลลัพธ์ซึ่งมีรูปแบบการเก็บข้อมูลอยู่หลากหลายลักษณะ อาทิเช่น Sets Lists Sorted Sets Strings และ Hashes เป็นต้น ซึ่งทางผู้จัดทำได้ทำการออกแบบรูปแบบการเก็บข้อมูลคู่คีย์ผลลัพธ์สำหรับข้อมูลจากตาราง โดยมีวิธีการและขั้นตอนดังต่อไปนี้

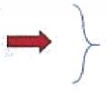
1. ดึงข้อมูลจากตารางมาพิจารณาทีละแถว
2. สร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes เพื่อเก็บข้อมูลแต่ละแถวของตาราง
3. ตั้งชื่อคีย์ ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยนำชื่อตารางระบุเป็นชื่อของคีย์และตามด้วย ":" จากนั้นตามด้วยระบุลำดับของแถวที่กำลังพิจารณาอยู่ ตั้งเป็นชื่อคีย์ของชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ดังกล่าว
4. ใส่ค่าข้อมูล Field ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยก่อนอื่นต้องพิจารณาก่อนว่าตารางที่กำลังพิจารณามีกี่สดมภ์หรือกี่แอททริบิวต์ โดยในขั้นตอนนี้จะทำการใส่ค่าของ Field ด้วยชื่อของแอททริบิวต์แต่ละตัวที่ระบุจากตารางข้อมูล
5. ใส่ค่าข้อมูลผลลัพธ์ลงบนชุดข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes ที่สร้างขึ้นมาจากขั้นตอนที่ 2 โดยจะทำการดึงข้อมูลจากตารางของแต่ละสดมภ์หรือแอททริบิวต์ โดยต้องทำการใส่ข้อมูลดังกล่าวต่อจาก Field ที่ได้ระบุค่าหรืออ้างอิงจากแอททริบิวต์หรือสดมภ์ของข้อมูลนั้น ๆ
6. ทำซ้ำในขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 ให้ครบทุกแถวของตารางข้อมูล

เพื่อสร้างความเข้าใจสำหรับวิธีการแปลงข้อมูลในรูปแบบที่ 2 มากยิ่งขึ้น ทางผู้จัดทำจึงได้นำเสนอวิธีการแปลงข้อมูลในรูปแบบที่ 1 นี้ผ่านตัวอย่างดังต่อไปนี้

ขั้นตอนที่ 1 ดึงข้อมูลจากตารางมาพิจารณาทีละแถว ซึ่งแสดงดังรูปที่ 3.10

Department Table

Department_number	Department_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resource
d001	Marketing
d004	Production
d008	Research
d006	Quality Management
d007	Sales

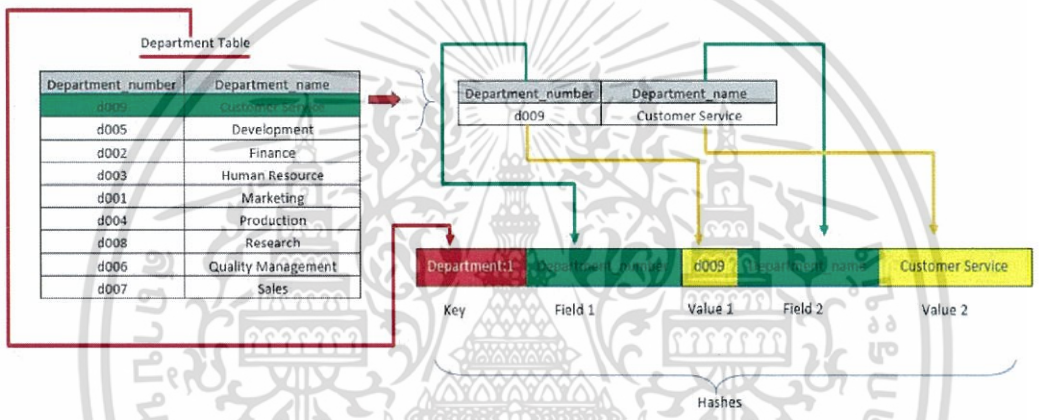


Department_number	Department_name
d009	Customer Service

รูปที่ 3.10 การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนที่ 2 และ 3 ทำการสร้างข้อมูลคู่คีย์ผลลัพธ์ชนิด Hashes เพื่อเก็บข้อมูลจากตารางแต่ละแถว โดยตั้งชื่อคีย์ข้อมูลจากชื่อตารางและตามด้วยเครื่องหมาย ":" และตามด้วยลำดับของแถวที่กำลังพิจารณาอยู่ จากนั้นในขั้นตอนที่ 4 ทำการสร้างค่าข้อมูล Field โดยต้องพิจารณาก่อนว่าตารางที่กำลังพิจารณามีกี่สดมภ์หรือกี่แอททริบิวต์ โดยใส่ชื่อของสดมภ์หรือชื่อแอททริบิวต์จากตารางข้อมูล ซึ่งพิจารณาใส่ข้อมูลจากสดมภ์ทางซ้ายก่อน จากนั้นจึงพิจารณาใส่ข้อมูลจากสดมภ์ถัดไปทางขวา และในขั้นตอนที่ 5 ทำการใส่ข้อมูลผลลัพธ์จากแถวที่กำลังพิจารณาอยู่ โดยต้องทำการใส่ข้อมูลดังกล่าวต่อจาก Field ที่ได้ระบุหรืออ้างอิงจากชื่อแอททริบิวต์หรือสดมภ์ของข้อมูลนั้น ๆ ตามลำดับ ดังแสดงในรูปที่ 3.11



รูปที่ 3.11 การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 2 3 4 และ 5

ในขั้นตอนที่ 6 ทำซ้ำในขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 ให้ครบทุกแถวของตารางข้อมูล ดังแสดงผลลัพธ์การแปลงข้อมูลในรูปแบบที่ 2 ดังรูปที่ 3.12

รูปที่ 3.12 การแปลงข้อมูลตารางเป็นรูปแบบชุดข้อมูลคู่คีย์ผลลัพธ์ขั้นตอนที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การเปรียบเทียบรูปแบบคำสั่งระหว่างฐานข้อมูลมาเรียดีบีและฐานข้อมูลเรดิส

ลักษณะรูปแบบคำสั่งระหว่างฐานข้อมูลมาเรียดีบี และฐานข้อมูลเรดิส มีโครงสร้างที่แตกต่างกันออกไปและฐานข้อมูลแต่ละชนิดมีวิธีการใช้งานคำสั่งที่แตกต่างกันออกไปเช่นกัน ดังนั้นการนำฐานข้อมูลทั้ง 2 ชนิดมาเปรียบเทียบหรือนำมาศึกษาในงานวิจัย จะต้องทำการศึกษาถึงลักษณะและรูปแบบการใช้งานคำสั่งต่าง ๆ ของฐานข้อมูลทั้ง 2 ชนิด โดยฐานข้อมูล มาเรียดีบี ใช้ภาษาเอสคิวแอลในการสั่งการประมวลผลต่าง ๆ ในระบบฐานข้อมูล ส่วนฐานข้อมูลชนิดใช้คีย์อ้างอิง ต่างมีรูปแบบโครงสร้างในการเก็บข้อมูลที่หลากหลายแตกต่างกันออกไป อาทิเช่น Strings Hashes Sets Lists และSorted Sets เป็นต้น และฐานข้อมูลเรดิส ก็มีรูปแบบคำสั่งในการเรียกใช้งานรูปแบบโครงสร้างดังกล่าวแตกต่างกัน โดยขึ้นอยู่กับนักพัฒนาว่าจะนำรูปแบบการเก็บข้อมูลแบบใดมาปรับใช้ในงานของตน โดยทางวิทยานิพนธ์ฉบับนี้ได้อ้างอิงถึงวิธีการของ [11] เพื่อเป็นแนวทางในทำงานวิจัยฉบับนี้

โดยรูปแบบคำสั่งระหว่างฐานข้อมูลมาเรียดีบี และฐานข้อมูลเรดิส ที่นำมาเปรียบเทียบประกอบด้วย คำสั่งสำหรับการอ่าน การลบ การแก้ไข และการอ่าน ซึ่งแสดงตัวอย่างการเปรียบเทียบในตารางที่ 3.3

ตารางที่ 3.3 ตัวอย่างการเปรียบเทียบคำสั่งสำหรับการอ่าน การลบ การแก้ไข และการอ่านของฐานข้อมูลมาเรียดีบี และฐานข้อมูลเรดิส

	มาเรียดีบี	เรดิส
การเขียน	INSERT INTO user(id, Name, Job) VALUES(“0”, “Dew”, “Engineer”)	$redis -> hmset(‘user:0’, ‘Name’, ‘Dew’ , ‘Job’ , ‘Engineer’); $redis -> zadd(‘user:Name:Dew’, 1, 0); $redis -> zadd(‘user:Job:Engineer’, 2, 0);

การลบ	DELETE FROM user WHERE Name = “Ken”	$name = $redis ->
		zrange(‘user:Name:Ken’,0,-1);
		for($i = 0; $i<count($name);$i++)
		{
		$redis->del(‘user:’.name[$i]);
		}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตัวอย่างการเปรียบเทียบคำสั่งสำหรับการอ่าน การลบ การแก้ไข และการอ่านของฐานข้อมูลมาเรียดีบี และฐานข้อมูลเรดิส (ต่อ)

	มาเรียดีบี	เรดิส
การแก้ไข	UPDATE user SET Name = “Meaw” ,Job = “Doctor” WHERE id = “0”	$name = $redis -> hmget(‘user:0’ , ‘Name’); $job = $redis -> hmget(‘user:0’ , ‘Job’); $redis -> del(‘user:Name:’.$name); $redis -> del(‘user:Job’.$job); $redis -> del(‘user:0’); $redis - > hmset(user:0, ‘Name’, ‘Meaw’ , ‘Job’ , ‘Doctor’); $redis -> zadd(‘user:Name:Meaw’, 1, 0); $redis -> zadd(‘user:Job:Doctor’, 2, 0);
การอ่าน	SELECT * FROM user WHERE id = “3”	$redis -> hmget(‘user:3’ , ‘Name’ , ‘Job’);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

การพัฒนาโปรแกรมทดสอบเปรียบเทียบความเร็วในการประมวลผลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง ซึ่งฐานข้อมูลเชิงสัมพันธ์ที่ใช้ในการทดลอง คือ มาเรียดีบี (MariaDB) และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่ใช้ในการทดลอง คือ เรดิส (Redis) โดยผู้จัดทำวิทยานิพนธ์ได้ทำการทดสอบเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผล โดยแบ่งการทดลองออกเป็น 2 ชุดการทดลอง ได้แก่

1. การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง
2. การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

การทดลองทั้ง 2 ชุดการทดลอง จะประกอบด้วยการทดลอง 4 ตัวดำเนินการ ได้แก่ การเขียน (Insert Operation) การลบ (Delete Operation) การแก้ไข (Update Operation) และการอ่าน (Select Operation) โดยบทนี้จะแสดงผลการทดลองการเปรียบเทียบประสิทธิภาพด้านความเร็วการประมวลผลของทั้ง 2 ชุดการทดลอง ระหว่างฐานข้อมูล มาเรียดีบี และฐานข้อมูล เรดิส และอภิปรายผลการทดลองทั้งหมด

4.1 ผลการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง

ทางผู้จัดทำเปรียบเทียบประสิทธิภาพการประมวลผลของฐานข้อมูลเชิงสัมพันธ์ มาเรียดีบี และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เรดิส เมื่อประมวลผลข้อมูลที่มีลักษณะเป็นโครงสร้างหรือเป็นข้อมูลที่มีความสัมพันธ์กัน โดยแบ่งการทดลองออกเป็น 4 ตัวดำเนินการ ได้แก่ การเขียน การลบ การแก้ไข การอ่าน ตามลำดับ เพื่อให้พิสูจน์ให้เห็นถึงจุดเด่นและจุดด้อยของฐานข้อมูลแต่ละแบบในแต่ละตัวดำเนินการ ทางผู้จัดทำได้ออกแบบการทดลองให้มีความหลากหลายลักษณะในแต่ละตัวดำเนินการ

4.1.1 ผลการทดลองด้านการเขียนข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง

ผลการทดลองการเปรียบเทียบความเร็วในการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างของฐานข้อมูลแต่ละแบบ โดยในการทดลองนี้ได้คำนึงถึงจำนวนข้อมูลในการเขียนข้อมูลเป็นสำคัญในการ

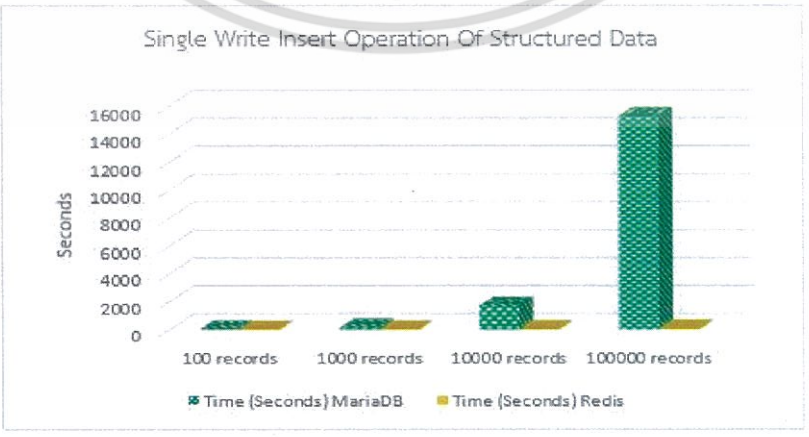
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาผลการทดลอง โดยทำการเขียนข้อมูลลงบนฐานข้อมูลแต่ละชนิด เป็นจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน 100,000 ระเบียน ตามลำดับ โดยการทดลองนี้ได้แบ่งการเขียนข้อมูลออกเป็น 2 รูปแบบ คือ การเขียนข้อมูลลงบนระบบฐานข้อมูลแบบที่ละคำสั่ง (Single Write) และการเขียนข้อมูลแบบรวมคำสั่ง (Multiple Write) ซึ่งสำหรับฐานข้อมูลเรดิส ได้ใช้ฟังก์ชันการเขียนข้อมูลพิเศษที่เรียกว่า Pipeline Feature ซึ่งฟังก์ชันพิเศษที่ทำให้สามารถรวมคำสั่งหลาย ๆ คำสั่งที่ต่อเนื่องกัน แล้วส่งให้ฐานข้อมูลเรดิส ทำงานเพียงครั้งเดียว โดยไม่ต้องรอให้เซิร์ฟเวอร์ตอบกลับการร้องขอของแต่ละคำสั่งก่อน ซึ่งเป็นฟังก์ชันการเขียนข้อมูลแบบรวมคำสั่งของฐานข้อมูลเรดิส โดยผลการทดลองได้แสดงในตารางที่ 4.1 และ 4.2 ตามลำดับ

ตารางที่ 4.1 ผลการทดลองการเขียนข้อมูลแบบที่ละคำสั่ง

Rows Number	Time (Seconds)	
	MariaDB	Redis
100 records	12.20271492	0.067361188
1000 records	141.6709618	0.495434284
10000 records	1698.276832	4.658929372
100000 records	15252.71206	58.84066243

ผลการทดลองในตารางที่ 4.1 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล (Rows Number) และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด (Seconds) ซึ่งแสดงดังรูปที่ 4.1



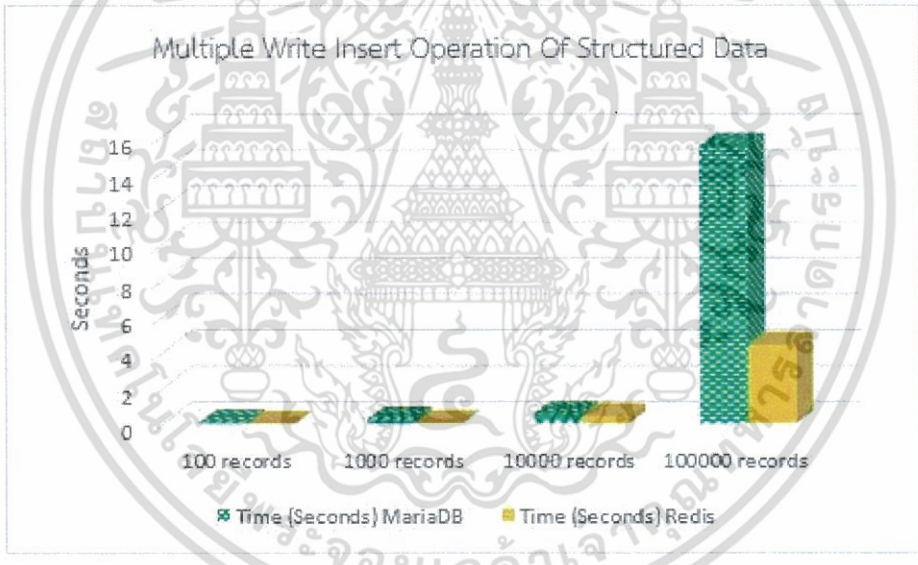
รูปที่ 4.1 ผลการทดลองการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างแบบทีละคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดลองการเขียนแบบรวมคำสั่ง

Rows Number	Time (Seconds)	
	MariaDB	Redis
100 records	0.119839668	0.005759282
1000 records	0.256996703	0.045402265
10000 records	0.560609105	0.404214001
100000 records	15.4682229	4.375085235

ผลการทดลองในตารางที่ 4.2 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟ เพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ซึ่งแสดงดังรูปที่ 4.2



รูปที่ 4.2 ผลการทดลองการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างแบบรวมคำสั่ง

ผลการทดลองการเขียนข้อมูลที่มีลักษณะเป็นโครงสร้างของทั้ง 2 รูปแบบ ดังแสดงในรูปที่ 4.1 และ 4.2 แสดงให้เห็นว่า เมื่อจำนวนข้อมูลเพิ่มสูงขึ้น ฐานข้อมูลเรดิส มีความเร็วในการประมวลผลเขียนข้อมูลได้เร็วกว่าฐานข้อมูลมาเรียดีบี เพราะฉะนั้นจากผลการทดลองนี้ สามารถสรุปได้ว่าฐานข้อมูลเรดิส สามารถรองรับการเขียนข้อมูลที่มีจำนวนมากได้ดีกว่าฐานข้อมูลมาเรียดีบี

4.1.2 ผลการทดลองด้านการลบข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง

ผลการทดลองความสามารถในการลบข้อมูลที่มีลักษณะเป็นโครงสร้าง โดยการทดลองการลบข้อมูลได้แบ่งการทดลองออกเป็น 2 ลักษณะ โดยการลบข้อมูลลักษณะที่ 1 จะคำนึงถึงปริมาณ

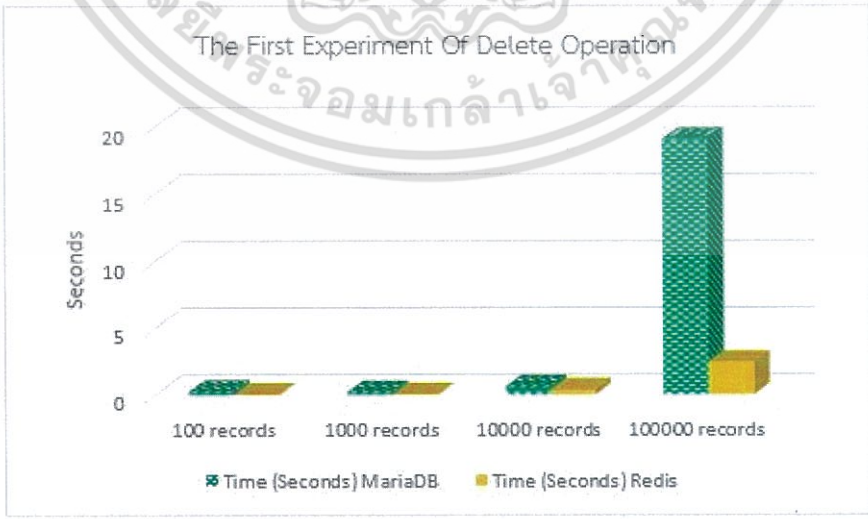
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเป็นสำคัญ ซึ่งจะสังเกตประสิทธิภาพการลบข้อมูลเมื่อปริมาณข้อมูลมีจำนวนเพิ่มขึ้น จากน้อยไปมาก โดยการลบข้อมูลลักษณะที่ 1 จะทำการลบข้อมูลจากระบบฐานข้อมูลทั้งสองชนิด โดยพิจารณาลบข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน และการลบข้อมูลในลักษณะที่ 2 เป็นการลบข้อมูลโดยคำนึงถึงเงื่อนไขเฉพาะเป็นสำคัญ หรือเรียกว่าลบข้อมูลจากเงื่อนไข ซึ่งทางผู้จัดทำได้ออกแบบการทดลองออกเป็นหลายลักษณะเพื่อเปรียบเทียบจุดเด่นและจุดด้อยของฐานข้อมูลแต่ละแบบ โดยผลการทดลองได้แสดงในตารางที่ 4.3 และ 4.4 ตามลำดับ

ตารางที่ 4.3 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

Rows Number	Time (Seconds)	
	MariaDB	Redis
100 records	0.256991768	0.024661231
1000 records	0.197009993	0.042733002
10000 records	0.616804573	0.357965302
100000 records	19.08710384	2.419591281

ผลการทดลองในตารางที่ 4.3 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ซึ่งแสดงดังรูปที่ 4.3



รูปที่ 4.3 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

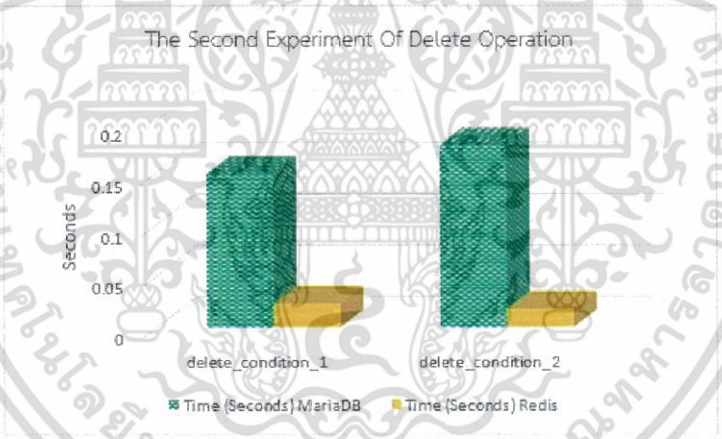
ตารางที่ 4.4 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2

Condition	Time (Seconds)	
	MariaDB	Redis
delete_condition_1	0.148897719	0.022788477
delete_condition_2	0.17667483	0.01755867

ผลการทดลองในตารางที่ 4.4 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ การลบข้อมูลในแต่ละเงื่อนไขและระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด (Seconds) ดังแสดงในรูปที่ 4.4 โดยเงื่อนไขที่ใช้ในการทดลองลบข้อมูลลักษณะที่ 2 มีรายละเอียดดังนี้

เงื่อนไขที่ 1 : ลบข้อมูลจากตาราง salaries ที่มี emp_no = '10004'

เงื่อนไขที่ 2 : ลบข้อมูลจากตาราง departments ที่มี dept_no = 'd009'



รูปที่ 4.4 ผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2

การพิจารณาผลการทดลองการลบข้อมูลที่มีลักษณะเป็นโครงสร้าง ทั้ง 2 ลักษณะ ที่แสดงดังตารางที่ 4.3 และ 4.4 ตามลำดับ เมื่อเพิ่มจำนวนข้อมูลการลบข้อมูลจากน้อยไปมาก จากผลการทดลองแสดงให้เห็นว่าฐานข้อมูลเรดิส ใช้เวลาในการประมวลผลลบข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบี และเมื่อลบข้อมูลโดยพิจารณาจากเงื่อนไข โดยเงื่อนไขที่ 1 คือการลบข้อมูลจากตาราง salaries เมื่อมี emp_no = 10004 และเงื่อนไขที่ 2 คือการลบข้อมูลจากตาราง departments เมื่อมี dept_no = d009 จากผลการทดลองแสดงให้เห็นว่าฐานข้อมูลเรดิส ใช้เวลาในการประมวลผลลบข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบี จากผลการทดลองการลบข้อมูลทั้ง 2 ลักษณะ ทำให้สามารถสรุปได้ว่า ฐานข้อมูลเรดิส มีประสิทธิภาพในการลบข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

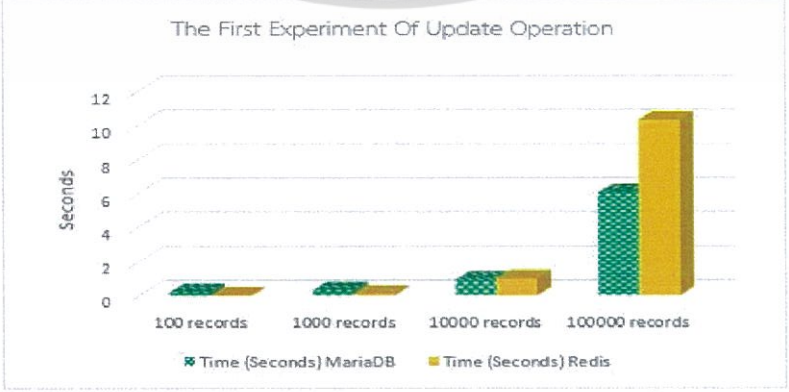
4.1.3 ผลการทดลองด้านการแก้ไขข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง

ผลการทดลองความสามารถในการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้าง โดยการทดลองการแก้ไขข้อมูลได้แบ่งการทดลองออกเป็น 2 ลักษณะ โดยการแก้ไขข้อมูลลักษณะที่ 1 จะคำนึงถึงปริมาณข้อมูลเป็นสำคัญ ซึ่งจะสังเกตประสิทธิภาพการแก้ไขข้อมูลเมื่อปริมาณข้อมูลมีจำนวนเพิ่มขึ้นจากน้อยไปมาก โดยการแก้ไขข้อมูลลักษณะที่ 1 จะทำการแก้ไขข้อมูลจากระบบฐานข้อมูลทั้งสองชนิด โดยพิจารณาแก้ไขข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน และการทดลองลักษณะที่ 2 คือการแก้ไขข้อมูลโดยพิจารณาอ้างอิงถึงข้อจำกัดหรือเงื่อนไขเฉพาะเป็นสำคัญ หรือแก้ไขข้อมูลตามเงื่อนไข ซึ่งทางผู้จัดทำได้ออกแบบการทดลองออกเป็นหลายลักษณะเพื่อเปรียบเทียบจุดเด่นและจุดด้อยของฐานข้อมูลแต่ละแบบ โดยผลการทดลองได้แสดงในตารางที่ 4.5 และ 4.6 ตามลำดับ

ตารางที่ 4.5 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

Rows Number	Time (Seconds)	
	MariaDB	Redis
100 records	0.21121798	0.014964742
1000 records	0.305551495	0.114862005
10000 records	0.931572338	0.988186665
100000 records	6.022390604	10.28571095

ผลการทดลองในตารางที่ 4.5 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ซึ่งแสดงดังรูปที่ 4.5



รูปที่ 4.5 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

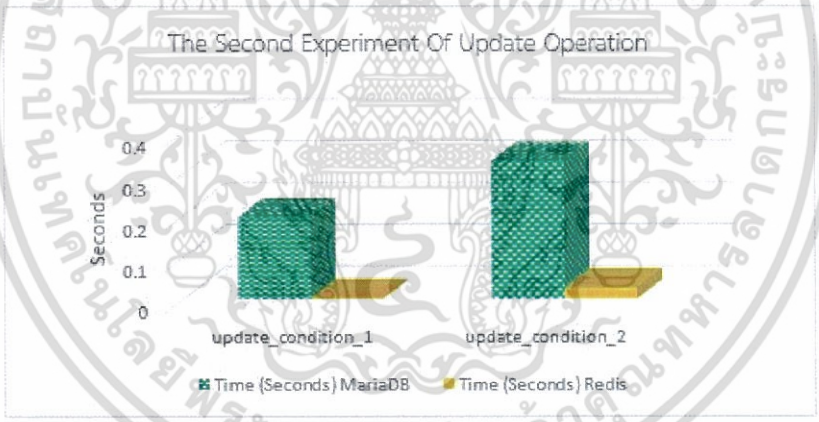
ตารางที่ 4.6 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2

Condition	Time (Seconds)	
	MariaDB	Redis
update_condition_1	0.196463993	0.002333549
update_condition_2	0.330927253	0.02656481

ผลการทดลองในตารางที่ 4.6 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟ เพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ การแก้ไขข้อมูลในแต่ละเงื่อนไขและระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ซึ่งแสดงดังรูปที่ 4.6 เงื่อนไขที่ใช้ในการทดลองแก้ไขข้อมูลลักษณะที่ 2 มีรายละเอียดดังนี้

เงื่อนไขที่ 1 : แก้ไขข้อมูลจากตาราง departments ที่มี dept_no = 'd001'

เงื่อนไขที่ 2 : แก้ไขข้อมูลจากตาราง titles ซึ่งทำการแก้ไขข้อมูล title และ to_date เมื่อมี emp_no = 10277 และมี from_date = 1985-06-16



รูปที่ 4.6 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2

ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะเป็นโครงสร้างทั้ง 2 ลักษณะ ซึ่งผลการทดลองได้แสดงดังตารางที่ 4.5 และ 4.6 จะแสดงให้เห็นว่าเมื่อทำการแก้ไขข้อมูลที่มีปริมาณ 100 ระเบียน และ1,000 ระเบียน ฐานข้อมูลเรดิส ใช้เวลาในการแก้ไขข้อมูลน้อยกว่าฐานข้อมูลมาเรีย แต่เมื่อเพิ่มจำนวนข้อมูลในการแก้ไขข้อมูลเป็น 10,000 ระเบียนและ100,000 ระเบียน ฐานข้อมูล มาเรียดีบี มีประสิทธิภาพการแก้ไขข้อมูลดีกว่าฐานข้อมูลเรดิส เพราะฉะนั้นจึงสามารถสรุปได้ว่าเมื่อปริมาณข้อมูลเพิ่มขึ้น ฐานข้อมูลมาเรียดีบี สามารถรองรับการแก้ไขข้อมูลได้ดีกว่าฐานข้อมูลเรดิส แต่ในการทดลองแก้ไขข้อมูลในลักษณะที่ 2 เมื่อแก้ไขข้อมูลโดยการพิจารณาถึงเงื่อนไข โดยเงื่อนไขการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ 1 คือ การแก้ไขข้อมูลจากตาราง departments เมื่อมี dept_no = d001 และเงื่อนไขการแก้ไขข้อมูลที่ 2 คือ การแก้ไขข้อมูลจากตาราง titles เมื่อมีค่า emp_no = 10277 และมีค่า from_date = 1985-06-16 ซึ่งจากผลการทดลองในตารางที่ 4.6 แสดงให้เห็นว่า ฐานข้อมูลเรดิส ใช้เวลาในการแก้ไขข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบี เมื่อแก้ไขข้อมูลโดยอ้างอิงจากเงื่อนไข ดังนั้นจึงสามารถสรุปได้ว่าฐานข้อมูลเรดิสจะมีประสิทธิภาพการแก้ไขข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี เมื่อพิจารณาการแก้ไขข้อมูลจากเงื่อนไขเฉพาะ

4.1.4 ผลการทดลองด้านการอ่านข้อมูลสำหรับข้อมูลที่มีลักษณะเป็นโครงสร้าง

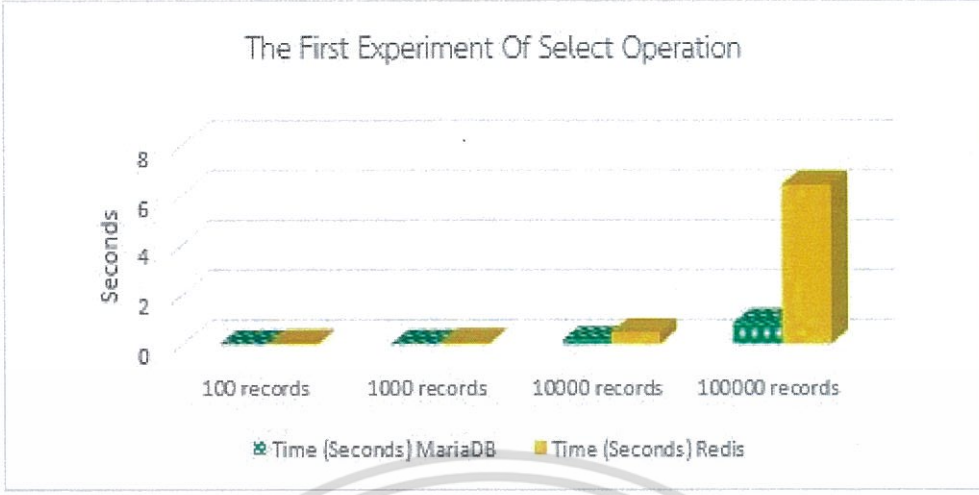
ผลการทดลองความสามารถในการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้าง โดยการทดลองการอ่านข้อมูลได้แบ่งการทดลองออกเป็น 2 ลักษณะ โดยลักษณะที่ 1 จะทำการอ่านข้อมูลจากระบบฐานข้อมูลทั้งสองชนิด โดยพิจารณาอ่านข้อมูลจำนวน 100 ระเบียน 1,000 ระเบียน 10,000 ระเบียน และ 100,000 ระเบียน ซึ่งการทดลองการแก้ไขข้อมูลในลักษณะที่ 1 คำนึงถึงปริมาณข้อมูลในการอ่านข้อมูลเป็นสำคัญในการพิจารณาผลการทดลอง และการทดลองลักษณะที่ 2 คือการอ่านข้อมูลโดยพิจารณาอ้างอิงถึงข้อจำกัดหรือเงื่อนไขเฉพาะ หรือเรียกว่าเป็นการอ่านข้อมูลที่มีเงื่อนไข โดยในการทดลองการอ่านข้อมูลลักษณะที่ 2 ได้แบ่งการอ่านข้อมูลที่มีเงื่อนไขออกเป็น 2 รูปแบบ คือ การอ่านข้อมูลแบบอย่างง่ายและการอ่านข้อมูลแบบซับซ้อน โดยการอ่านข้อมูลแบบอย่างง่าย คือการอ่านข้อมูลที่มีเงื่อนไขจากข้อมูล 1 ตาราง ส่วนการอ่านข้อมูลแบบซับซ้อน คือการอ่านข้อมูลที่มีเงื่อนไขจากตารางข้อมูลมากกว่า 1 ตารางขึ้นไป ซึ่งเป็นการอ่านข้อมูลที่มีความสัมพันธ์กัน ซึ่งทางผู้จัดทำได้ออกการการทดลองออกเป็นหลายลักษณะเพื่อเปรียบเทียบจุดเด่นและจุดด้อยของฐานข้อมูลแต่ละแบบ โดยผลการทดลองได้แสดงในตารางที่ 4.7 และ 4.8 ตามลำดับ

ผลการทดลองในตารางที่ 4.7 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ดังแสดงในรูปที่ 4.7

ตารางที่ 4.7 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

Rows Number	Time (Seconds)	
	MariaDB	Redis
100 records	0.003638449	0.028316855
1000 records	0.01338382	0.065310951
10000 records	0.157204213	0.458379236
100000 records	0.844943895	6.368828368

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 1

ตารางที่ 4.8 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างลักษณะที่ 2

Select Condition	Time (Seconds)	
	MariaDB	Redis
Q1(Simple query)	3.83952951	15.27403824
Q2(Simple query)	0.03593229	0.002695107
Q3(Complex query)	19.7769188	23.37216868
Q4(Complex query)	0.020542	0.009211254
Q5(Complex query)	0.02030049	0.009487688
Q6(Complex query)	0.010794612	0.009497349
Q7(Complex query)	1.438579011	0.738623595

ผลการทดลองในตารางที่ 4.8 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ การอ่านข้อมูลในแต่ละเงื่อนไขและระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด โดยผลการทดลองในการอ่านข้อมูลในลักษณะที่ 2 นี้ ทางผู้จัดทำได้แบ่งรูปกราฟออกเป็น 2 รูป ได้แก่ กราฟแสดงผลการทดลองการอ่านข้อมูลแบบมีเงื่อนไขในรูปแบบอย่างง่าย และกราฟแสดงผลการทดลองการอ่านข้อมูลแบบมีเงื่อนไขในรูปแบบซับซ้อน ดังรูปที่ 4.8 และ 4.9 โดยการทดลองในการอ่านข้อมูลในลักษณะที่ 2 นี้จะประกอบด้วยเงื่อนไขการอ่านข้อมูลตามรายละเอียดดังต่อไปนี้

1. อ่านข้อมูลส่วนตัวของพนักงาน ที่มีเพศเป็นผู้ชาย (อย่างง่าย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อ่านข้อมูลจากตาราง dept_emp ที่มีรหัสพนักงาน = 10021 (อย่างง่าย)

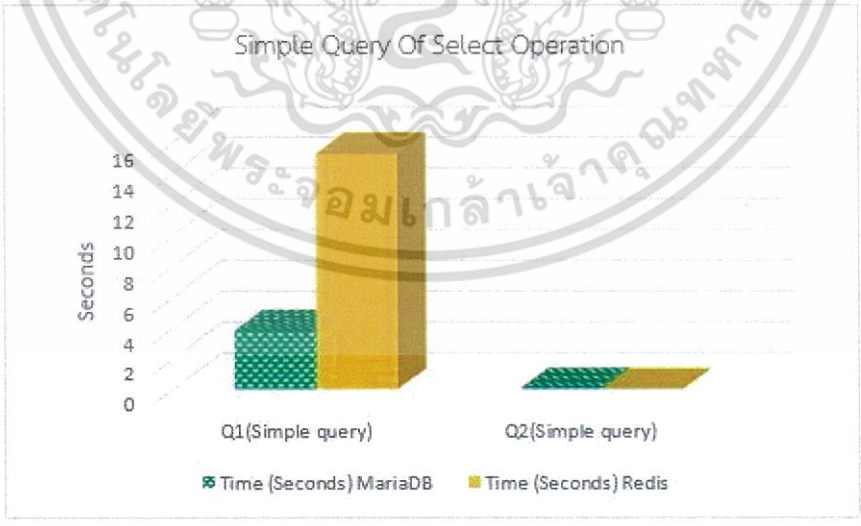
3. อ่านข้อมูลจากตาราง dept_emp และตาราง departments เพื่อหาว่าพนักงานแต่ละคนอยู่ในแผนกไหนบ้าง และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)

4. อ่านข้อมูลจากตาราง departments และ dept_manager เพื่อหาว่า พนักงานที่มีตำแหน่ง ระดับพนักงาน อยู่ในแผนกไหนบ้าง และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)

5. อ่านข้อมูลส่วนตัวของพนักงานและเป็นพนักงานที่เป็นระดับผู้จัดการ และหาว่าพนักงานคนนั้นอยู่ในแผนกไหน ซึ่งทำการอ่านข้อมูลจาก 3 ตาราง คือ ตาราง employees ตาราง dept_manager และตาราง departments และใช้รูปแบบการอ่านข้อมูลแบบจอยบนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)

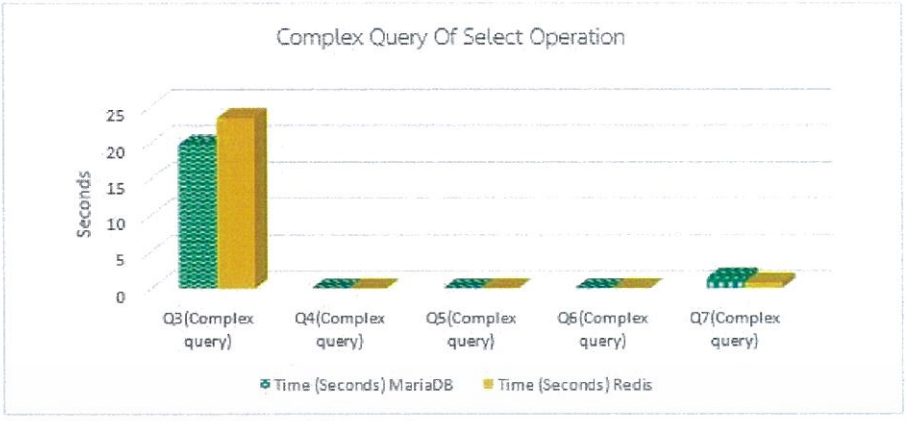
6. อ่านข้อมูลส่วนตัวของพนักงาน ซึ่งมีตำแหน่งเป็นระดับผู้จัดการ และมีรหัส dept_no = “d001” ซึ่งทำการอ่านข้อมูลจากตาราง employees และตาราง dept_manager โดยใช้รูปแบบการอ่านข้อมูลแบบซับคิวรี่บนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)

7. อ่านข้อมูลส่วนตัวของพนักงาน ซึ่งมีเงินเดือนน้อยกว่าหรือเท่ากับ 50,000 บาท ซึ่งทำการอ่านข้อมูลจากตาราง employees และตาราง salaries โดยใช้รูปแบบการอ่านข้อมูลแบบซับคิวรี่บนระบบฐานข้อมูลมาเรียดีบี (ซับซ้อน)



รูปที่ 4.8 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในรูปแบบอย่างง่ายลักษณะที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในรูปแบบซับซ้อนลักษณะที่ 2

จากผลการทดลองการอ่านข้อมูลที่แสดงในตารางที่ 4.7 และ 4.8 พบว่าการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในลักษณะที่ 1 ฐานข้อมูลมาเรียดีบี สามารถรองรับการอ่านข้อมูลที่มีปริมาณมากได้ดีกว่าฐานข้อมูลเรดิส ซึ่งแสดงให้เห็นดังรูปกราฟรูปที่ 4.7 เมื่อปริมาณข้อมูลเพิ่มขึ้นฐานข้อมูลมาเรียดีบีใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลเรดิส ส่วนผลการทดลองการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างในลักษณะที่ 2 ทางผู้จัดทำได้แบ่งการทดลองนี้ออกเป็น 2 รูปแบบ คือการอ่านข้อมูลอย่างง่ายและการอ่านข้อมูลแบบซับซ้อน โดยรูปกราฟรูปที่ 4.8 แสดงผลการทดลองการอ่านข้อมูลอย่างง่ายและรูปกราฟรูปที่ 4.9 แสดงผลการทดลองการอ่านข้อมูลแบบซับซ้อน จากรูปที่ 4.8 ในการอ่านข้อมูลในเงื่อนไขที่ 1 หรือ Q1 เป็นการอ่านข้อมูลส่วนตัวของพนักงานที่มีเพศเป็นเพศชาย ซึ่งเป็นการอ่านข้อมูลที่แสดงผลลัพธ์มากกว่าหนึ่งผลลัพธ์จากข้อมูล 1 ตาราง ซึ่งผลการทดลองแสดงให้เห็นว่าฐานข้อมูลมาเรียดีบี ใช้เวลาในการประมวลผลอ่านข้อมูลน้อยกว่าฐานข้อมูลเรดิส ดังแสดงผลลัพธ์ใน Q1 แต่เมื่อทำการอ่านข้อมูลในเงื่อนไขที่ 2 หรือ Q2 ซึ่งเป็นการอ่านข้อมูลจากตาราง dept_emp ที่มีค่า emp_no = 10021 ซึ่งเป็นการอ่านข้อมูลที่แสดงผลลัพธ์เพียงค่าเดียวจากฐานข้อมูล 1 ตาราง โดยผลการทดลองแสดงให้เห็นว่าฐานข้อมูลเรดิสใช้เวลาในการประมวลผลอ่านข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบี ดังแสดงผลลัพธ์ใน Q2 และผลการทดลองการอ่านข้อมูลแบบซับซ้อน จากรูปที่ 4.9 โดยผลการทดลองการอ่านข้อมูลในเงื่อนไขที่ 3 หรือ Q3 ทางผู้จัดทำได้ทำการอ่านข้อมูลจากข้อมูล 2 ตาราง ได้แก่ตาราง dept_emp และตาราง departments โดยตัวดำเนินการแบบจอย (Join Operation) ถูกใช้ในการอ่านข้อมูลสำหรับฐานข้อมูลมาเรียดีบี โดยผลการทดลองการอ่านข้อมูลในเงื่อนไขที่ 3 นี้แสดงให้เห็นว่า ฐานข้อมูลมาเรียดีบีใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลเรดิส และการทดลองการอ่านข้อมูลในเงื่อนไขที่ 4 หรือ Q4 ทางผู้จัดทำได้ทำการอ่านข้อมูลจากข้อมูล 2 ตาราง ได้แก่ตาราง departments และตาราง dept_manager โดยตัวดำเนินการแบบจอย ถูกใช้ในการอ่านข้อมูลสำหรับฐานข้อมูลมาเรียดีบี ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองการอ่านข้อมูลในเงื่อนไขที่ 4 นี้แสดงให้เห็นว่าฐานข้อมูลเรดิส ใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลมาเรียดีบี แต่เมื่อทำการอ่านข้อมูลในเงื่อนไขที่ 5 หรือ Q5 โดยอ่านข้อมูลจากตารางข้อมูล 3 ตาราง ได้แก่ตาราง employees ตาราง dept_manager และตาราง departments โดยตัวดำเนินการแบบจอยถูกใช้ในการอ่านข้อมูลสำหรับฐานข้อมูลมาเรียดีบี ซึ่งผลการทดลองการอ่านข้อมูลในเงื่อนไขที่ 5 นี้แสดงให้เห็นว่าฐานข้อมูลมาเรียดีบี ใช้เวลาในการประมวลผลมากกว่าฐานข้อมูลเรดิส และในการทดลองอ่านข้อมูลในเงื่อนไขที่ 6 และ 7 (Q6 และ Q7) เป็นการอ่านข้อมูลที่มีความสัมพันธ์กัน โดยตัวดำเนินการแบบซับคิวรี่ (Subquery Operation) ได้ถูกนำมาใช้ในการอ่านข้อมูลสำหรับฐานข้อมูลมาเรียดีบี โดยการอ่านข้อมูลในเงื่อนไขที่ 6 เป็นการอ่านข้อมูลส่วนตัวของพนักงาน ที่มีตำแหน่งเป็นระดับผู้จัดการ ที่มีรหัส dept_no = d001 โดยทำการอ่านข้อมูลจากข้อมูล 2 ตาราง ได้แก่ตาราง employees และตาราง dept_manager และการอ่านข้อมูลในเงื่อนไขที่ 7 เป็นการอ่านข้อมูลส่วนตัวของพนักงาน ซึ่งมีเงินเดือนน้อยกว่าหรือเท่ากับ 50,000 บาท ซึ่งทำการอ่านข้อมูลจากข้อมูล 2 ตาราง ได้แก่ตาราง employees และตาราง salaries จากผลการทดลองในตารางที่ 4.8 แสดงให้เห็นว่าฐานข้อมูลมาเรียดีบี ใช้เวลาในการประมวลผลมากกว่าฐานข้อมูลเรดิส ทั้งในส่วนของการอ่านข้อมูลในเงื่อนไขที่ 6 และ 7 ตามลำดับ จากการวิเคราะห์ผลการทดลองการอ่านข้อมูลที่มีเงื่อนไขซับซ้อน จะเห็นได้ว่าฐานข้อมูลเรดิสสามารถรองรับการอ่านข้อมูลที่มีรูปแบบซับซ้อนได้ดีกว่าฐานข้อมูลมาเรียดีบี จากการวิเคราะห์การอ่านข้อมูลในลักษณะที่ 2 ทั้งหมด โดยรวมฐานข้อมูลเรดิสใช้เวลาประมวลผลอ่านข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบี ถึงแม้มีการอ่านข้อมูลบางเงื่อนไขที่ใช้เวลาอ่านข้อมูลมากกว่าฐานข้อมูลมาเรียดีบีก็ตาม แต่ระยะเวลาในการอ่านข้อมูลเงื่อนไขดังกล่าวยังถือว่าช้ากว่าไม่มากนัก ดังนั้นจึงสรุปได้ว่าฐานข้อมูลเรดิสมีประสิทธิภาพในการอ่านข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี เมื่ออ่านข้อมูลโดยอ้างอิงจากเงื่อนไข และจากผลการทดลองทั้งหมดของการอ่านข้อมูลที่มีลักษณะเป็นโครงสร้างสามารถสรุปได้ว่าฐานข้อมูลมาเรียดีบี สามารถรองรับการอ่านข้อมูลที่มีลักษณะที่เป็นโครงสร้างและมีปริมาณมากได้ดีกว่าฐานข้อมูลเรดิส แต่เมื่อทำการอ่านข้อมูลแบบมีเงื่อนไขที่มีความซับซ้อนมากขึ้นทำให้ประสิทธิภาพการอ่านข้อมูลของฐานข้อมูลมาเรียดีบี ลดลงมากเมื่อเปรียบเทียบกับฐานข้อมูลเรดิส

4.2 ผลการทดลองเปรียบเทียบความเร็วการประมวลผลชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

ทางผู้จัดทำเปรียบเทียบประสิทธิภาพการประมวลผลของฐานข้อมูลเชิงสัมพันธ์ มาเรียดีบี และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เรดิส เมื่อประมวลผลข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่และเป็นข้อมูลที่ปริมาณมหาศาล ที่จำนวน 1,844,244 ระเบียน เพื่อเปรียบเทียบประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการประมวลข้อมูลขนาดใหญ่ของฐานข้อมูลแต่ละแบบ โดยแบ่งการทดลองออกเป็น 4 ตัวดำเนินการ ได้แก่ การเขียน การลบ การแก้ไขและการอ่าน ตามลำดับ ซึ่งแสดงผลการทดลองดังตารางที่ 4.9 นอกจากนี้ทางผู้จัดทำได้ทำการทดลอง การเขียน การลบ การแก้ไข และการอ่านสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ใน 3 ระดับข้อมูล ที่มีปริมาณข้อมูล 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ โดยทางผู้จัดทำต้องการสังเกตถึงประสิทธิภาพของฐานข้อมูลแต่ละชนิดเมื่อปริมาณข้อมูลเพิ่มขึ้นจากน้อยไปมาก ซึ่งแสดงผลการทดลองการเขียนข้อมูลทั้ง 3 ระดับข้อมูลในตารางที่ 4.10 และแสดงผลการทดลองการลบข้อมูลทั้ง 3 ระดับข้อมูลในตารางที่ 4.11 และแสดงผลการทดลองการแก้ไขข้อมูลทั้ง 3 ระดับข้อมูลในตารางที่ 4.12 และแสดงผลการทดลองการอ่านข้อมูลทั้ง 3 ระดับข้อมูลในตารางที่ 4.13

ตารางที่ 4.9 ผลการทดลองการประมวลผลสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

Operation	Time (Seconds)	
	MariaDB	Redis
Insert Operation	370.3730142	147.2472775
Delete Operation	252.3812836	0.326949644
Update Operation	249.2234956	326.2293523
Select Operation	18.26119201	7.88261447

ผลการทดลองในตารางที่ 4.9 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ ตัวดำเนินการ และระยะเวลาการประมวลผลของฐานข้อมูลมาเรียดีบีกับฐานข้อมูลเรดิส ดังแสดงในรูปที่ 4.10

รูปที่ 4.10 ผลการทดลองการประมวลผลสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองการเปรียบเทียบความเร็วการประมวลผลสำหรับข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ ที่มีจำนวน 1,844,244 ระเบียน ซึ่งแสดงในตารางที่ 4.9 และรูปที่ 4.10 ของฐานข้อมูลแต่ละแบบ โดยผลการทดลองด้านการเขียนข้อมูลแสดงให้เห็นว่า ฐานข้อมูลเรดิสใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลมาเรียดีบี และผลการทดลองด้านการลบข้อมูล แสดงให้เห็นว่า ฐานข้อมูลเรดิส ใช้เวลาในการลบข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบีอยู่ถึงหลายเท่า ส่วนผลการทดลองแก้ไขข้อมูล แสดงให้เห็นว่าฐานข้อมูลมาเรียดีบีใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลเรดิส และผลการทดลองการอ่านข้อมูล แสดงให้เห็นว่าฐานข้อมูลเรดิส ใช้เวลาในการประมวลผลน้อยกว่าฐานข้อมูลมาเรียดีบี ดังนั้นจึงสามารถสรุปได้ว่า ฐานข้อมูลเรดิสสามารถรองรับการเขียน การลบ และการอ่านข้อมูลลักษณะของข้อมูลขนาดใหญ่ที่มีปริมาณมหาศาล ได้ดีกว่าฐานข้อมูลมาเรียดีบี แต่ฐานข้อมูลเรดิสมีจุดอ่อนในเรื่องของการแก้ไขข้อมูล

ตารางที่ 4.10 ผลการทดลองการเขียนข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

Insert Operation	Time (Seconds)	
	MariaDB	Redis
500,000 records	90.43698444	33.11877679
1,000,000 records	225.2594855	92.03567512
1,500,000 records	321.4238482	116.5900505

ผลการทดลองในตารางที่ 4.10 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ดังแสดงในรูปที่ 4.11

รูปที่ 4.11 ผลการทดลองการเขียนข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

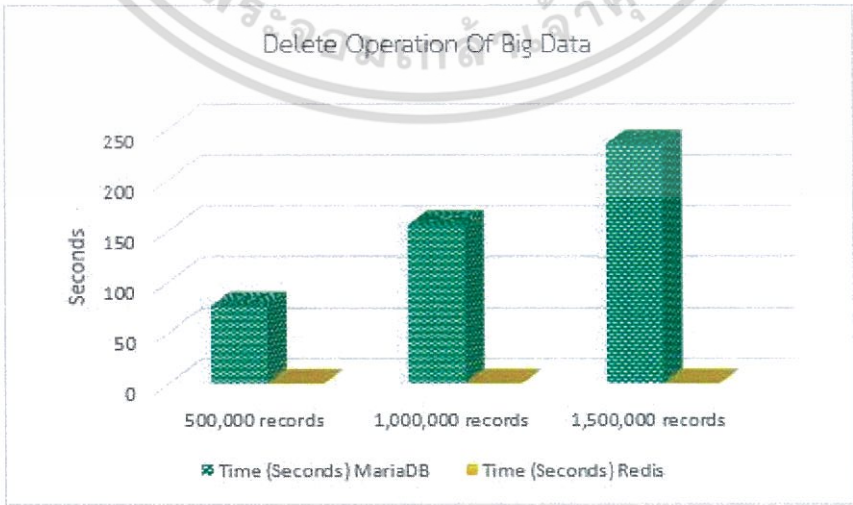
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองเขียนข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล ตั้งแต่ข้อมูลจำนวน 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ จากผลการทดลองที่แสดงในตารางที่ 4.10 และรูปที่ 4.11 แสดงให้เห็นว่า เมื่อปริมาณข้อมูลเพิ่มขึ้นจากน้อยไปมาก ประสิทธิภาพการประมวลผลของฐานข้อมูลแต่ละชนิดได้แปรผันตามขนาดของข้อมูลที่เพิ่มขึ้น โดยฐานข้อมูลเรดิสจะใช้เวลาในการเขียนข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบีในทุกระดับของข้อมูล ดังนั้นจากการวิเคราะห์ผลการทดลองจึงสรุปได้ว่า ฐานข้อมูลเรดิสมีประสิทธิภาพด้านการเขียนข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี

ตารางที่ 4.11 ผลการทดลองการลบข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

Delete Operation	Time (Seconds)	
	MariaDB	Redis
500,000 records	76.60455544	0.031117772
1,000,000 records	156.1352069	0.062314701
1,500,000 records	235.2039281	0.075696754

ผลการทดลองในตารางที่ 4.11 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ดังแสดงในรูปที่ 4.12



รูปที่ 4.12 ผลการทดลองการลบข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

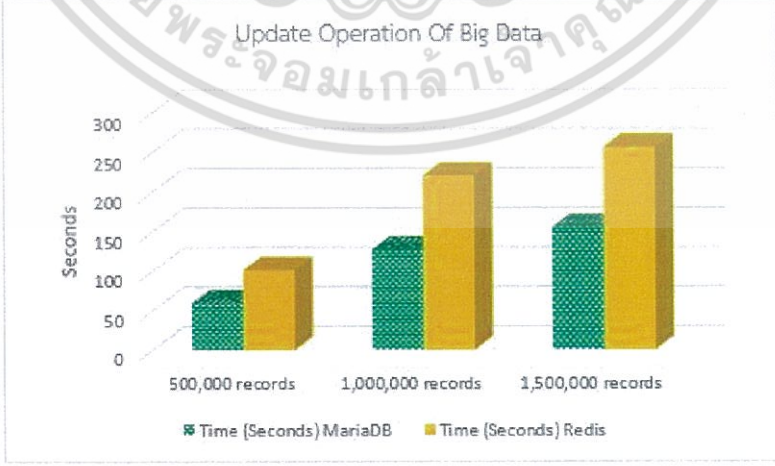
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองลบข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล ตั้งแต่ข้อมูลจำนวน 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ จากผลการทดลองที่แสดงในตารางที่ 4.11 และรูปที่ 4.12 แสดงให้เห็นว่า เมื่อปริมาณข้อมูลเพิ่มขึ้นจากน้อยไปมาก ประสิทธิภาพการประมวลผลของฐานข้อมูลแต่ละชนิดได้แปรผันตามขนาดของข้อมูลที่เพิ่มขึ้น โดยฐานข้อมูลเรดิสจะใช้เวลาในการลบข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบีในทุกระดับของข้อมูล ดังนั้นจากการวิเคราะห์ผลการทดลองจึงสรุปได้ว่า ฐานข้อมูลเรดิสมีประสิทธิภาพด้านการลบข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี

ตารางที่ 4.12 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

Update Operation	Time (Seconds)	
	MariaDB	Redis
500,000 records	59.35604932	100.9921289
1,000,000 records	126.8730642	220.0577953
1,500,000 records	154.8052206	256.0516865

ผลการทดลองในตารางที่ 4.12 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ดังแสดงในรูปที่ 4.13



รูปที่ 4.13 ผลการทดลองการแก้ไขข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

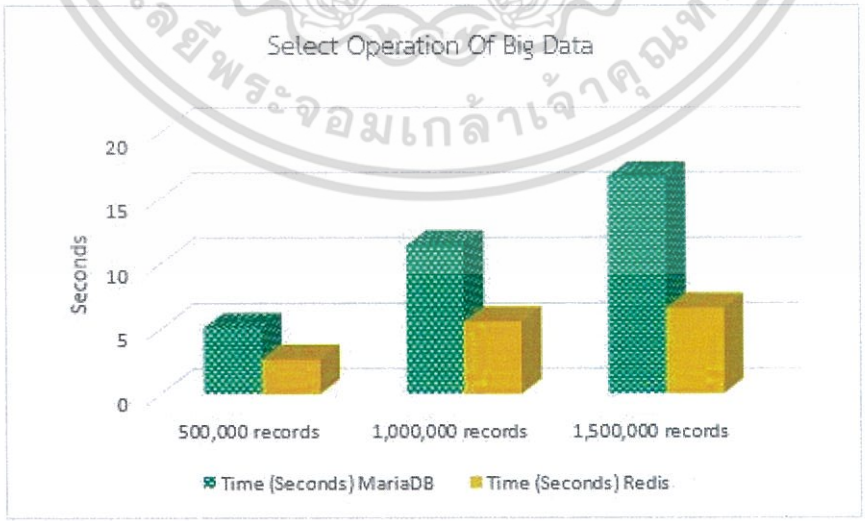
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองแก้ไขข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล ตั้งแต่ข้อมูลจำนวน 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ จากผลการทดลองที่แสดงในตารางที่ 4.12 และรูปที่ 4.13 แสดงให้เห็นว่า เมื่อปริมาณข้อมูลเพิ่มขึ้นจากน้อยไปมาก ประสิทธิภาพการประมวลผลของฐานข้อมูลแต่ละชนิดได้แปรผันตามขนาดของข้อมูลที่เพิ่มขึ้น โดยฐานข้อมูลมาเรียดีบีจะใช้เวลาในการแก้ไขข้อมูลน้อยกว่าฐานข้อมูลเรดิสในทุกระดับของข้อมูล ดังนั้นจากการวิเคราะห์ผลการทดลองจึงสรุปได้ว่า ฐานข้อมูลมาเรียดีบีมีประสิทธิภาพด้านการแก้ไขข้อมูลที่ดีกว่าฐานข้อมูลเรดิส

ตารางที่ 4.13 ผลการทดลองการอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

Select Operation	Time (Seconds)	
	MariaDB	Redis
500,000 records	5.101649785	2.67245371
1,000,000 records	11.4102314	5.552646422
1,500,000 records	16.6514657	6.588314533

ผลการทดลองในตารางที่ 4.13 ทางผู้จัดทำได้นำผลการทดลองมาเขียนในรูปแบบของกราฟเพื่อให้เห็นภาพรวมที่ชัดเจนมากขึ้น โดยแบ่งเป็น 2 แกน ได้แก่ จำนวนของข้อมูล และระยะเวลาในการประมวลผลของฐานข้อมูลแต่ละชนิด ดังแสดงในรูปที่ 4.14



รูปที่ 4.14 ผลการทดลองการอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ 3 ระดับข้อมูล ตั้งแต่ข้อมูลจำนวน 500,000 ระเบียน 1,000,000 ระเบียน และ 1,500,000 ระเบียน ตามลำดับ จากผลการทดลองที่แสดงในตารางที่ 4.13 และรูปที่ 4.14 แสดงให้เห็นว่า เมื่อปริมาณข้อมูลเพิ่มขึ้นจากน้อยไปมาก ประสิทธิภาพการประมวลผลของฐานข้อมูลแต่ละชนิดได้แปรผันตามขนาดของข้อมูลที่เพิ่มขึ้น โดยฐานข้อมูลเรดิสจะใช้เวลาในการอ่านข้อมูลน้อยกว่าฐานข้อมูลมาเรียดีบีในทุกระดับของข้อมูล ดังนั้นจากการวิเคราะห์ผลการทดลองจึงสรุปได้ว่า ฐานข้อมูลเรดิสมีประสิทธิภาพด้านการอ่านข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี

การวิเคราะห์ผลการทดลองการเขียน การลบ การแก้ไขและการอ่าน ของชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ ทั้ง 3 ระดับ จะเห็นได้ว่าประสิทธิภาพการประมวลผลข้อมูลของฐานข้อมูลแต่ละชนิดจะแปรผันตามขนาดของข้อมูลที่เพิ่มขึ้นจากน้อยไปมาก โดยผลการทดลองยังคงอยู่ในแนวโน้มเดิม คือฐานข้อมูลเรดิสจะมีประสิทธิภาพด้านการเขียน การลบ และการอ่านข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี แต่จะมีจุดอ่อนด้านการแก้ไขข้อมูล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

งานวิจัยในวิทยานิพนธ์ฉบับนี้เป็นงานวิจัยเชิงทดลอง (Experimental Research) ซึ่งมีวัตถุประสงค์ศึกษาเปรียบเทียบประสิทธิภาพด้านความเร็วในการประมวลผลข้อมูลระหว่างระบบฐานข้อมูลเชิงสัมพันธ์และระบบฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง เพื่อช่วยเป็นข้อบ่งชี้ถึงความเหมาะสมในการเลือกใช้งานระบบฐานข้อมูล โดยการทดลองในงานวิจัยฉบับนี้แบ่งการทดลองออกเป็น 2 การทดลอง ได้แก่ การทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะเป็นโครงสร้าง และการทดลองเปรียบเทียบความเร็วการประมวลผลสำหรับชุดข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่ โดยแต่ละการทดลองแบ่งออกเป็น 4 ตัวดำเนินการ คือ การเขียน การลบ การแก้ไขและการอ่านข้อมูล ซึ่งทุกการทดลองถูกทดลองบนชุดข้อมูลเดียวกันและทดลองในสภาวะแวดล้อมที่เหมือนกัน โดยฐานข้อมูลเชิงสัมพันธ์ที่นำมาทำการทดลอง คือ มาเรียดีบี (MariaDB) เวอร์ชัน 10.1.13-MariaDB และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงที่นำมาทำการทดลองคือ เรดิส (Redis) เวอร์ชัน 3.2.100. และงานวิจัยฉบับนี้ยังได้นำเสนอกระบวนการแปลงข้อมูลที่มีลักษณะเป็นโครงสร้างตารางไปเก็บในรูปแบบข้อมูลคู่คีย์ผลลัพธ์บนฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยพิจารณาอยู่ 2 ปัจจัยคือ ข้อมูลโครงสร้างและความสัมพันธ์ของตารางข้อมูลและข้อมูลที่ระบุอยู่ในตาราง

จากการวิเคราะห์และอภิปรายผลการทดลองทั้งหมดสามารถสรุปได้ว่า ทางด้านการเขียนข้อมูล จากการวิเคราะห์ผลการทดลองจะเห็นได้ว่า ฐานข้อมูลเรดิส มีประสิทธิภาพด้านการเขียนข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี จากการพิจารณาทั้ง 2 การทดลอง และทางด้านการลบข้อมูล จากการวิเคราะห์ผลการทดลองจะเห็นได้ว่า ฐานข้อมูลเรดิสก็มีประสิทธิภาพด้านการลบข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบีอีกเช่นกัน แต่ในส่วนของการแก้ไขข้อมูล จากการวิเคราะห์ผลการทดลองทั้ง 2 การทดลอง จะเห็นได้ว่า เมื่อทำการแก้ไขข้อมูลในปริมาณมากขึ้น ฐานข้อมูลมาเรียดีบีมีประสิทธิภาพในการแก้ไขข้อมูลที่ดีกว่าฐานข้อมูลเรดิส หรือกล่าวได้ว่าสามารถรองการการแก้ไขข้อมูลที่มีปริมาณมากได้ดีกว่าฐานข้อมูลเรดิส และการวิเคราะห์ผลการทดลองด้านการอ่านข้อมูล จากการทดลองที่ 1 เมื่ออ่านข้อมูลที่มีลักษณะเป็นโครงสร้าง เมื่อจำนวนข้อมูลในการอ่านข้อมูลเพิ่มขึ้น จากผลการทดลองแสดงให้เห็นว่าฐานข้อมูลมาเรียดีบีใช้เวลาในการอ่านข้อมูลน้อยกว่าฐานข้อมูลเรดิส เนื่องจากลักษณะของชุดข้อมูลที่มีลักษณะเป็นโครงสร้างและปริมาณข้อมูลที่ยังไม่ถือว่าเป็นปริมาณข้อมูลที่มหาศาล จึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ฐานข้อมูลมาเรียดีบีมีประสิทธิภาพในการอ่านข้อมูลที่ดีกว่าฐานข้อมูลเรดิส แต่ในการอ่านข้อมูลในการทดลองที่ 2 เมื่อทำการอ่านข้อมูลที่มีลักษณะของข้อมูลขนาดใหญ่และข้อมูลมีปริมาณที่มหาศาล จากผลการทดลองจะแสดงให้เห็นว่า ฐานข้อมูลเรดิส มีประสิทธิภาพในการอ่านข้อมูลที่ดีกว่าฐานข้อมูลมาเรียดีบี และจากการวิเคราะห์ผลการทดลองการอ่านข้อมูลที่มีเงื่อนไขที่ซับซ้อนในการทดลองที่ 1 จากผลการทดลองจะแสดงให้เห็นว่า ฐานข้อมูลเรดิส สามารถรองรับการอ่านข้อมูลที่มีความซับซ้อนได้ดี ซึ่งตรงกันข้ามกับฐานข้อมูลมาเรียดีบี เนื่องจากเมื่อมีการอ่านข้อมูลที่มีความซับซ้อนมากขึ้นประสิทธิภาพการอ่านข้อมูลของฐานข้อมูลมาเรียดีบีจะลดลง

จากการวิเคราะห์ผลการทดลอง จะเห็นได้ว่าฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง (เรดิส) ไม่ได้มีประสิทธิภาพดีกว่าฐานข้อมูลเชิงสัมพันธ์ (มาเรียดีบี) ในทุกการทดลอง ซึ่งแต่ละฐานข้อมูลแต่ละชนิดต่างมีจุดเด่นและจุดด้อยที่แตกต่างกัน ซึ่งฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิงมีจุดเด่นในด้านการเขียน การลบ การอ่านข้อมูลที่มีปริมาณมหาศาล และรองรับการอ่านข้อมูลที่มีรูปแบบซับซ้อนได้ดี แต่มีจุดอ่อนในด้านการแก้ไขข้อมูล ดังนั้นจากการวิเคราะห์ผลการทดลอง จึงสรุปว่าฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง (เรดิส) จึงเหมาะสำหรับแอปพลิเคชันที่ต้องการรองรับการประมวลผลข้อมูลที่มีปริมาณมหาศาล ที่ไม่เน้นในเรื่องการแก้ไขข้อมูลหรือเป็นแอปพลิเคชันที่มีอัตราการแก้ไขข้อมูลน้อยมาก อาทิเช่น แอปพลิเคชันที่รองรับการเก็บข้อมูลจำพวกเซ็นเซอร์ต่าง ๆ หรือเป็นระบบที่ใช้เก็บข้อมูลเฉพาะทางด้านประวัติศาสตร์หรือข้อมูลทางสถิติ หรือระบบรับข้อมูลพิกัดทางภูมิศาสร์ เป็นต้น และเหมาะกับแอปพลิเคชันที่รองรับการอ่านข้อมูลลักษณะที่ซับซ้อนได้ดี อาทิเช่น แอปพลิเคชันที่ใช้จัดการข้อมูลด้านโซเชียล เน็ตเวิร์ค เป็นต้น

5.2 ข้อเสนอแนะ

1. วิทยานิพนธ์ฉบับนี้มุ่งเน้นศึกษาการเปรียบเทียบประสิทธิภาพการประมวลผลระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง โดยทำการทดลองและประมวลผลอยู่บนเครื่องเดียว ดังนั้นถ้าหากทำการทดลองเปรียบเทียบประสิทธิภาพการประมวลแบบกระจาย ก็จะทำให้สามารถบ่งชี้ถึงประสิทธิภาพการประมวลผลในสภาวะอื่น ๆ ให้ครอบครุมมากยิ่งขึ้น

2. วิทยานิพนธ์ฉบับนี้มุ่งเน้นศึกษาการเปรียบเทียบประสิทธิภาพการประมวลผลระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลแบบไม่สัมพันธ์ชนิดใช้คีย์อ้างอิง ดังนั้นหากนำฐานข้อมูลแบบไม่สัมพันธ์ชนิดอื่น ๆ มาทำการทดลอง อาจจะทำให้เห็นความสัมพันธ์ของจุดเด่นและจุดด้อยของฐานข้อมูลแบบไม่สัมพันธ์แต่ละชนิดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Zachary P, Scott P. and Susan V. V. "Comparing NoSQL MongoDB to an SQL DB." **In Proceedings of the 51st ACM Southeast Conference**, 2013, April 4-6, p. 5.

[2] Zhu Wei-ping, Li Ming-Xin and Chen Huan "Using MongoDB to Implement Textbook Management System instead of MySQL" **In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on,** September 2011, pp.303-305.

[3] Alexandru B., Florin R. and Laura I. A. "MongoDB vs Oracle database comparison" **In Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on,** November 2012, pp. 330-335.

[4] Yishan l. and Sathiamoorthy M."A performance comparison of SQL and NoSQL databases" **In Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on,** October 2013, pp. 15-19.

[5] Cornelia G., Robert G., George P. and Andrada O. "A comparative study: MongoDB vs. MySQL" **In Engineering of Modern Electric Systems (EMES), 2015 13th International Conference on,** July 2015, pp. 1-6.

[6] Min-Gyue J., Seon-A Y., Jayon B. and Yong-Lak C. "A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment" **In Database Theory and Application (DTA), 2015 8th International Conference on,** March 2016, pp. 14-17.

[7] Satyadhyan C., Anoop G. and Ankita K. "Comparison of Relational Database with Document-Oriented Database(MongoDB) for Big Data Applications" **In Advanced Software Engineering & Its Applications (ASEA), 2015 8th International Conference on,** March 2016, pp. 41-47.

[8] Azhi F., Bilal R. and Twana S. "Comparative study of relational and non-relations database performances using Oracle and MongoDB systems" **In International Journal of Computer Engineering and Technology (IJCET)** ,Volume 5, Issue 11, November (2014), pp. 11-22.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง(ต่อ)

[9] Seyyed H. A., Mehdi R., Milad M. and Nasser G. "Performance evaluation of SQL and MongoDB databases for big e-commerce data" **In Computer Science and Software Engineering (CSSE), 2015 International Symposium on**, January 2016, pp. 1-7.

[10] Garima J. and Arun P. A. "Comparative analysis of Relational and Graph databases," **In IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719** Vol. 3, Issue 8 (August. 2013), ||V2|| PP 25-27

[11] Josh Sherman, "**MySQL and Redis Command Equivalents,**" [Online]. Available : https://joshtronic.com/2013/07/29/mysql-and-redis-command-equivalents/,2013

[12] Canonical Group Ltd., "**Sample database with test suite 1.0.6 employees-db-1.0.6,**" [Online]. Available : https://launchpad.net/test-db/employees-db-1/1.0.6 , 2009

[13] Pragati P. S., Saumya G. and Anil K. "Analysis of Various NoSql Database" **Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on**, January 2016, pp. 539-544.

[14] Vinoo Das, "**Learning Redis**" First published: June 2015, Birmingham: Packt Publishing Ltd., 2015

[15] Redis.io, "**An introduction to Redis data types and abstractions**", [Online], Available : https://redis.io/topics/data-types-intro, 2017

[16] Redis.io, "**SDIFF Key**", [Online], Available : https://redis.io/commands/sdiff, 2017

[17] Redis.io, "**SINTER Key**", [Online], Available : https://redis.io/commands/sinter, 2017

[18] Redis.io, "**SUNION Key**", [Online], Available : https://redis.io/commands/sunion ,2017

[19] Redis.io, **SPOP Key**, [Online], Available : https://redis.io/commands/spop, 2017

[20] solid IT, "**DB-Engines Ranking**", [Online], Available : http://db-engines.com/en/ranking , 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง(ต่อ)

[21] Wikipedia, "**Key value database**", [Online], Available : https://upload.wikimedia.org/wikipedia/commons/5/5b/KeyValue.PNG , 2016

[22] KDnuggets, "**Top NoSQL Database Engines**", [Online], Available : http://www.kdnuggets.com/wp-content/uploads/key-value.png ,2016

[23] Satyajit Pual, "Distributed Datastores – let 's take a look under the hood", [Online], Available : http://satyajit-paul.blogspot.com/2014_05_01_archive.html ,2014

[24] couchbase.com, "**Comparing document-oriented and relational data**", [Online], Available : https://developer.couchbase.com/documentation/server/3.x/developer/dev-guide-3.0/compare-docs-vs-relational.html, 2017

[25] Daniel Nogel, "**Graph based recommendation engine for Shopware**", [Online], Available : https://developers.shopware.com/blog/2015/10/14/graph-based-recommendation-engine-for-shopware/, 2015

[26] tutorrialspoint, "**LEARN CASSANDRA : Data model**", [Online], Available : https://www.tutorialspoint.com/cassandra/cassandra_data_model.htm, 2017

[27] Microsoft, "**GeoLife GPS Trajectories**" , [Online], Available : https://www.microsoft.com/en-us/download/details.aspx?id=52367&from=https%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fdownloads%2Fb16d359d-d164-469e-9fd4-daa38f2b2e13%2F, 2012

[28] Wikipedia,"**MariaDB**", [Online], Available : https://en.wikipedia.org/wiki/MariaDB ,2017

[29] ThaiCreate.Com Team, "**รู้จัก MariaDB คืออะไร และเปรียบเรียบระหว่าง MariaDB กับ MySQL**", [Online], Available : http://www.thaicreate.com/tutorial/mariadb-mysql.html ,2013

[30] April Reeve, "**Big Data Architectures – NoSQL Use Cases for Key Value Databases**", [Online], Available : https://infocus.emc.com/april_reeve/big-data-architectures-nosql-use-cases-for-key-value-databases/ ,2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง(ต่อ)

[31] โอภาส เอี่ยมสิริวงศ์ , **ระบบฐานข้อมูล** , กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2551

[32] ประกายมาศ ศรีสุขทักษิณ, "**การเปรียบเทียบความเร็วในการประมวลผลระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลไม่สัมพันธ์แบบเอกสาร**" สารนิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2556

[33] Wikipedia, "**Key-value database**", [Online], Available : https://en.wikipedia.org/wiki/Key-value_database, 2017

[34] Redis.io, **Introduction to Redis**, [Online], Available : https://redis.io/topics/introduction, 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ผลงานทางวิชาการที่ได้รับการตีพิมพ์ในการประชุมทางวิชาการ

1. Wittawat Puangsaijai and Sutheera Puntheeranurak **"A Comparative Study of Relational Database and Key-Value Database for Big Data Applications"**, The 2017 International Electrical Engineering Congress (iEECON2017), Pattaya, Thailand, March 8-10, 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Comparative Study of Relational Database and Key-Value Database for Big Data Applications

Wittawat Puangsaijai
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
57601304@kmitl.ac.th

Sutheera Puntheeranurak
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
sutheera.pu@kmitl.ac.th

Abstract— **Nowadays, Demands of web scale are in increasing and growing rapidly. Mobile applications, web technologies, social media always generates unstructured data that had lead to the advent of various NoSQL databases. Therefore, Big data applications are necessary to have an efficient technology to collect these data. However, a relational database is the traditional database that always uses in many applications and still has more valuable to play a significant role in the current information system. The main characteristics of NoSQL databases are schema-free, no relationship, no need to join as a relational database. The business organization expects that NoSQL database has better performance than a relational database. In This paper, we aim to compare the performance of Redis, which is a key-value database, one kind of NoSQL database, and MariaDB, which is a popular relational database. We designed a set of experiments with a large amount of data and compared the efficiency of the insert, update, delete and select transactions from various aspects on the same dataset. We measure the processing time of each transaction to evaluate the comparison. The results have shown that Redis has better runtime performance for insert, delete, update transaction under a specific condition or complex queries. MariaDB still is good for some conditions especially when we have a small data. Our study can help to choose a database that will be suitable for the real world applications because relational databases and NoSQL databases have different strengths and weakness.**

Keywords—NoSQL database; Redis; Key-value database, Big data applications, MariaDB.

I. INTRODUCTION

In the past, many applications widely used relational databases to collect information. The relational database resembled the basic structure in the form of a table which contained a row and column. It could be ordered through the SQL query language to insert, delete, update, and query data. By the way, a technology that can scale, perform and deliver continuous availability needs for supporting online applications. Thus, relational databases have struggled to keep up with the wave of modernization.

The most viable database option for online web and mobile applications have to support the amount of data that increases tremendously as well as the type of data with more variety such as unstructured data, structured data, and semi-structured data. Moreover, it is necessary to have a database system that can be processed with large amounts of data quickly and with flexibility situations. Although the Relational database is famous, it may not be compatible with the growing requirements that require more performance system. To overcome this problem, NoSQL, which is a non-relational database, is introduced. Therefore, there are many discussions in the database world to replace the relational database system with the NoSQL database system. Currently, NoSQL database has been more popular in the current information system. However, there is a question for the database designer to choose NoSQL database or relational database to store information in their systems. They have information that NoSQL database is useful for unstructured data to store a large quantity of data. In fact, when we need to collect structured data and to store a large amount of data, NoSQL database may be good enough when not compare to the relational database because the relational database can collect data in the form of a table and can store data that is structured as well.

Also, NoSQL databases provide high flexibility. They do not have a fixed database schema. They can store information in a variety of formats. We choose Redis as representatives of NoSQL databases because it is a robust open source database with a growing community developing and supporting it.

In this paper, we aim to compare the performance between Redis as NoSQL database which is the key-value type and MariaDB as a relational database. We investigate which situation that one is better and measure the performance from processing time. We experiment on insert operation, delete operation, update operation and select operation in the same dataset.

The paper is organized as follows. In section 2, we discuss related work.We show our experimental setup in section 3 and describe the results of our experiments in section 4. We describe the conclusion and future work in section 5.

II. RELATED WORKS

In recent year, there are many researchers offered the comparison between a non-relational database and relational database because they would like to find the suitable database for their work and there are many NoSQL databases that have advantages and disadvantages. Many papers compare the efficiency between MongoDB and a relational database. Such

978-1-5090-4666-9/17/$31.00 ©2017 IEEE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

as [1],[2] compared the effectiveness between MongoDB and SQL server. In [1] experimented in 3 operations that consisted of the insert, update, select operation. The result of running an experiment showed that MongoDB had better runtime performance for insert, update and simple query. SQL server performed better when updating and querying by non-key attributes, as well as for aggregate queries. In [2] experimented in 5 operations that consisted of the insert, update, delete, select and general queries. They tested on the same dataset for a common e-commerce schema. The results showed that MongoDB performed better than SQL server for most operations except aggregate function queries. Moreover, some research, In [3] compared MongoDB with MySQL. In [3] divided the experiment into four operations that consisted of the insert, update, delete and select operation. They experimented in forum application. The results of the experiment showed MongoDB had better performance than MySQL. In [4] compared performance of querying. They divided querying into two categories that consisted of the simple query and complex query. Their results showed MongoDB had better performance than Oracle in the simple query, but Oracle had better performance than MongoDB in a complex query. In [5] experimented to compare the performance of graph database (Neo4J) and relational database (MySQL). Their results showed that Neo4J was faster than MySQL in querying the results. Many researchers use MongoDB to be representative of NoSQL database because it established longer than other databases. However, it can show only that MongoDB, one kind of document databases, is outperformed than a relational database in many case studies. There are still have many NoSQL databases that are useful.

In this paper, we focus on the comparison between Redis as NoSQL database and MariaDB as a relational database in the same environment.We do our experiments in four operations that consist of an insert operation, an update operation, delete operation and select operation. The several queries performed on Redis database which equivalent queries in SQL language have been carried out on MariaDB. We refer to the method of [6] for uses as guidelines in the coding of our experiment.

III. Experimental Setup

For our experimental setup, we installed both Redis and MariaDB on the same machine that is running an Intel Core i5-2450M 2.5 GHz with 8 GB of DDR3 memory. This experiment was developed on Windows operating system and was coded in PHP framework and using Xampp Server to simulate a web server. For Redis, we used a library which named predis for connecting between PHP framework and Redis database and used the latest version 3.2.100. We used MariaDB version 10.1.13 in our experiments.

A. Data Model Schema

We download data from sample database with test suite 1.0.6 [7]. The data model schema used for the experiments as shown in Fig. 1. The data model consisted of 6 tables/objects representing the employee, dept_manager, department, salary, title, and dept_emp. The data of the employee table contains 300,024 records. The dept_manager table contains 24 records. The data of department table includes 9 records. The salary table contains 2,844,047 records. The title table and the dept_emp table contains 443,308 records and 331,603 records, respectively.

B. Experimental Design

We use PHP language to develop the experiments that run under four scales of data as 100, 1,000, 10,000, 100,000 records. We divided into four operations.

1) Insert Operation: To add data, we used Salary table in this operation. We split into two styles of inserting. First, we add one by one command to the databases, called single write. Second, we combine all of the insert commands to execute at once, called multiple inserts. In Redis database, we used the pipelining feature as multiple inserts. The pipelining feature is the way that we can send multiple commands to execute at once without waiting for the server replies. Therefore, Pipelining feature is very necessary for Redis to evaluate the performance of insertion operation.

2) Delete Operation: We separate into two experiments. First, we delete data with defined scales from salary table at one time. Second, we delete data by considering the specific conditions.

3) Update Operation: We separate into two experiments as update salary table with four scales of data and update data with the specific conditions.

4) Select Operation: First, we query data from salary table that have the results as four mentioned scales with no conditions. Second, we separate queries into two categories which consist of the simple query and complex query. The simple query involves selecting data from only one table. The complex query involves selecting data from multiple tables and selecting data in nested queries. The following list summarizes querying in the second experiment.

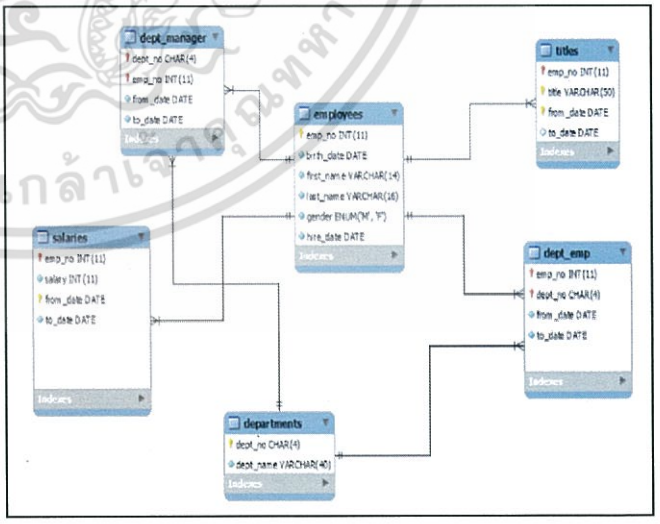


Fig. 1. Data model schema of our experiment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Q1: Read personal data of the employees from the employee table with gender = "M" (simple query)
- Q2: Read data from dept_emp table with emp_no = "10021" (simple query)
- Q3: Read data from dept_emp table and dept_name table. To find out which department of each employee. We use join operation on MariaDB query. (complex query)
- Q4: Read data from department table and dept_manager table. To find out which department of each employee who is a manager. We use join operation on MariaDB query. (complex query)
- Q5: Read data from employee table, dept_manager table and department table. To find out personal data and departments' name of each employee who is a manager. We use join operation on MariaDB query. (complex query)
- Q6: Read data from employee table and dept_manager table. To find out the personal data of employees who is a manager in dept_no = "d001". We use a subquery in MariaDB query. (complex query)
- Q7: Read data from employee table and salary table. To find out Personal data of employees whose salary is less than or equal to 50,000 baht. We use a subquery in MariaDB query. (complex query)

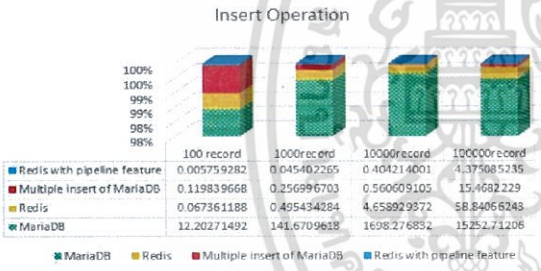


Fig. 2. The results of the insert operation.

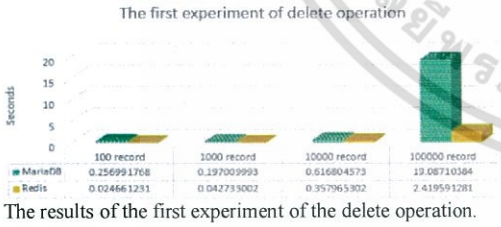


Fig. 3. The results of the first experiment of the delete operation.

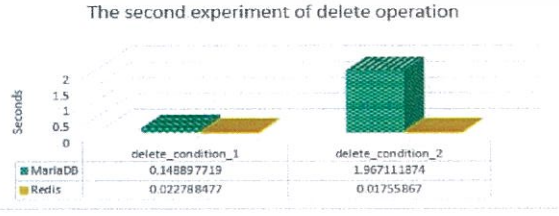


Fig. 4. The results of the second experiment of the delete operation.

We ran ten times for each test and calculated the average performance time. In our experiment, we used the PHP function microtime which is used to measure the duration of processing.

IV. Experimental Results

A. Insert Operation

When inserting the data into both databases with four scales of 100, 1,000, 10,000, 100,000 records, respectively. The results show that Redis is more outperform than MariaDB for both styles of insertion as shown in Fig. 2.

B. Delete Operation

In delete operation, we separate into two experiments.

1) The first test of delete operation

We execute the first test of a delete operation in both databases with four scales of data. Thus, the results in Fig. Three show that Redis is faster than MariaDB in this case.

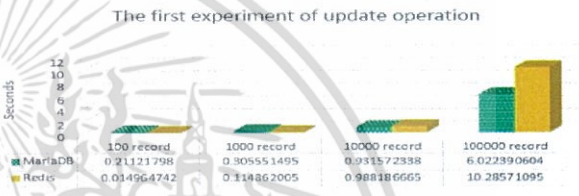


Fig. 5. The results of the first experiment of the update operation.

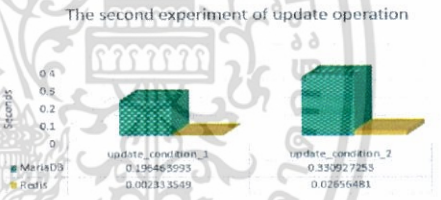


Fig. 6. The results of the second experiment of the update operation.

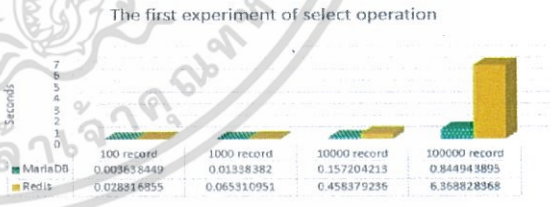


Fig. 7. The results of the first experiment of select operation.

Fig. 8. The results of the second experiment of select operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) The second test of delete operation

The second test of a delete operation, we remove the data by defining the specific conditions. The following list summarizes the conditions that we used in this analysis.

delete_condition_1: Delete data from salary table with **emp_no = "10004"**

delete_condition_2: Delete data from department table **with dept_no = "d009"**

In Fig. 4, the results show that Redis has better performance than MariaDB in this experiment. Therefore, we can conclude that for all styles delete operation; Redis is more efficiency than MariaDB.

C. Update Operation

In update operation, we separate into two experiments.

1) The first experiment of update operation

We applied update operation in Redis and MariaDB. In Fig. Five can show that MariaDB is faster than Redis for update operation without any specific conditions even though data is very huge.

2) The second experiment of update operation

We modified the data by considering the specific conditions and used the following conditions list in our second experiment.

update_condition_1: Update data from department table **with dept_no = "d001"**

update_condition_2: Update data from title table with emp_no = 10277 and from_date = 1985-06-16

For the results in Fig. 6, it can show that Redis is outperformed more than MariaDB when we updated with specific conditions.

D. Select Operation

In select operation, we separate into two experiments.

1) The first experiment of select operation

We use simple query without any specific conditions on Redis and MariaDB. In Fig. 7, it can show that MariaDB provides lower execution times than Redis. Therefore, we can conclude that MariaDB has more performance than Redis when we use selection operation even on the larger data.

2) The second experiment of select operation

We divide into two categories which consist of the simple query and complex query.

The performance of the simple query, Q1, indicates that MariaDB has better than Redis when we refer to the multiple results from the single table. However, Redis is better than MariaDB when we query a single result from the table by using Q2. In the case of complex query, join operation, or subquery operation is used to query data in MariaDB. When we run Q4, Q5, Q6, and Q7 on Redis and MariaDB, the results is shown in Fig. 8. It can demonstrate that the performance of MariaDB will decrease when we execute the complex query as query data from more than one table and query data by using a subquery operation. MariaDB spent the time to process query more than Redis. However, only Q3 in complex query's experiment shows that MariaDB is slightly faster than Redis.

V. Conclusion

The replacing a relational database with NoSQL database need to study and find the suitable solution for a particular application, especially big data applications. Therefore, it is necessary to consider feature and every operation before developers make a decision which database should be useful in big data applications. It depends on their requirements.

From the experimental results of the comparison between Redis, NoSQL database as a key-value database, and MariaDB, relational database, can show that NoSQL database does not get more performance than the relational database in every operation. Redis has better runtime performance of insert operation, delete operation, update operation especially updating by using specific conditions, and select operation in querying single result of the simple query as well as querying data of complex query. Because Redis has many advantages such as flexible, no fixed schema, and easy to use. Furthermore, the data model of MariaDB is more complex. Thus, it decreased querying performance. However, Redis also has disadvantages in regular querying and updating that applied in an extensive data. Therefore, we should use Redis in the form of a temporary database to enhance the performance of database systems.

In future works, we plan to enhance by involving several NoSQL database systems to improve the performance of an application by evaluating Redis and MariaDB as a distributed database.

References

[1] Z. Parker, S. Poe and S.V. Vrbsky, "Comparing NoSQL MongoDB to an SQL DB," In Proceedings of the 51st ACM Southeast Conference, p. 5. ACM, 2013.

[2] S.H. Aboutorabi, M. Rezapour, M. Moradi and N. Ghadiri, "Performance evaluation of SQL and MongoDB databases for big e-commerce data," In Computer Science and Software Engineering (CSSE), 2015 International Symposium on, pp. 1-7. IEEE. 2015.

[3] C. Győrödi, R. Győrödi, G. Pecherle and A Olah, "A comparative study: MongoDB vs. MySQL," In Engineering of Modern Electric Systems (EMES), 2015 13th International Conference on, pp. 1-6. IEEE, 2015.

[4] A. Faraj, B. Rashid and T. Shareef, "Comparative study of relational and non-relations database performances using Oracle and MongoDB systems," In International Journal of Computer Engineering and Technology (IJCET) Volume 5, Issue 11, November (2014), pp. 11-22.

[5] G. Jaiswal and A.P. Agrawal, "Comparative Analysis of Relational and Graph Databases," In IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 8 (August. 2013), ||V2|| PP 25-27

[6] Josh Sherman, "MySQL and Redis Command Equivalents," Available in https://joshtronic.com/2013/07/29/mysql-and-redis-command-equivalents/.

[7] Canonical Group Ltd., "Sample database with test suite 1.0.6 "employees-db-1.0.6," Available in https://launchpad.net/test-db/employees-db-1/1.0.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นายวิทธวัช พวงสายใจ

วัน เดือน ปีเกิด 31 ตุลาคม 2534

ที่อยู่ 71 หมู่ 5 ต. ลวงเหนือ อ. ดอยสะเก็ด จ. เชียงใหม่ 50220

ประวัติการศึกษา 2541 – 2553 โรงเรียนดาราวิทยาลัย

2556 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ความชำนาญเฉพาะด้าน 1. พีเอชพี จาวาและซีโปรแกรมมิ่ง

2. ออกแบบพัฒนาระบบเชิงวัตถุและเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้