

การศึกษาเปรียบเทียบประสิทธิภาพการสืบค้นข้อมูลระหว่าง
ฐานข้อมูลเชิงสัมพันธ์กับฐานข้อมูล NoSQL สำหรับระบบ
สารสนเทศ กรณีศึกษาตัวอย่างชุดข้อมูลครุภัณฑ์ของสำนักบริการ
เทคโนโลยีสารสนเทศ

Comparative Study of Query Performance between
Relational Database and NoSQL Database
for Information System: A Case Study of the Asset
Database of Information Technology Service Center

เยาวภา จรัสสันติจิต^{1*}
Yaowapa Jaratsantijit^{1*}

Received: 11 February 2024

Revised: 15 December 2024

Accepted: 19 February 2025

บทคัดย่อ

การวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาแนวทางการจัดเก็บข้อมูลของฐานข้อมูล NoSQL และเปรียบเทียบประสิทธิภาพในการสืบค้นข้อมูลครุภัณฑ์ระหว่างฐานข้อมูลเชิงสัมพันธ์กับฐานข้อมูล NoSQL สำหรับระบบสารสนเทศ กรณีศึกษาตัวอย่างชุดข้อมูลครุภัณฑ์ของสำนักบริการเทคโนโลยีสารสนเทศมหาวิทยาลัยเชียงใหม่ โดยได้เลือกใช้ฐานข้อมูลเชิงสัมพันธ์คือ MySQL ซึ่งมีการจัดเก็บข้อมูลเป็นแบบตารางที่ประกอบด้วยแถวและคอลัมน์ เลือกใช้ฐานข้อมูล NoSQL คือ MongoDB ซึ่งมีการจัดเก็บข้อมูลในรูปแบบคีย์-ค่าที่ประกอบด้วย Field-Value และใช้ภาษา PHP ในการพัฒนาเว็บแอปพลิเคชันสำหรับทดสอบ ผู้วิจัยได้ออกแบบคำสั่งและพัฒนาเว็บแอปพลิเคชันเพื่อใช้ทดสอบประสิทธิภาพการสืบค้นระหว่างฐานข้อมูลทั้งสองประเภท ผลการทดสอบจากการศึกษาเบื้องต้นพบว่าการสืบค้นข้อมูลครุภัณฑ์ด้วยฐานข้อมูลเชิงสัมพันธ์มีประสิทธิภาพความเร็วในการสืบค้นที่เร็วกว่าการสืบค้นด้วยฐานข้อมูล NoSQL และผลลัพธ์ที่ได้จากการสืบค้นข้อมูลจากฐานข้อมูลทั้งสองประเภทต่างให้ผลลัพธ์ที่ถูกต้องแม่นยำและตรงกัน 100% สาเหตุที่ทำให้ประสิทธิภาพการสืบค้นข้อมูล

¹ สำนักบริการเทคโนโลยีสารสนเทศ มหาวิทยาลัยเชียงใหม่ เชียงใหม่ ประเทศไทย 50200

¹ Information Technology Service Center, Chiang Mai University, Chiang Mai, Thailand 50200

* ผู้นิพนธ์ประสานงาน (Corresponding author) e-mail: yaowapa.j@cmu.ac.th

ด้วยฐานข้อมูล NoSQL ค้นหาค้นหาข้อมูลได้ช้ากว่าเนื่องจากโครงสร้างการจัดเก็บข้อมูลครุภัณฑ์ในฐานข้อมูล NoSQL ซึ่งเมื่อถูกแปลงมาจากฐานข้อมูลเชิงสัมพันธ์แล้ว จะถูกจัดเก็บอยู่ในรูปแบบของ Referenced Document ที่มีการเก็บข้อมูลในคอลเลคชันที่แยกกัน การค้นหาจึงต้องใช้ฟังก์ชันในการเชื่อมโยงข้อมูลซึ่งมีผลต่อเวลาที่ใช้ในการประมวลผลที่มากขึ้นส่งผลให้ค้นหาข้อมูลได้ช้าลง และจากการศึกษาเพิ่มเติมเพื่อยืนยันว่าฐานข้อมูล NoSQL สามารถอ่านหรือเขียนข้อมูลขนาดใหญ่ได้รวดเร็วกว่าฐานข้อมูลเชิงสัมพันธ์จึงได้ทำการทดสอบเพิ่มโดยใช้ฐานข้อมูลตัวอย่างซึ่งประกอบด้วย ข้อมูลพนักงานและเงินเดือนพนักงานที่มีปริมาณข้อมูลมากกว่าข้อมูลครุภัณฑ์ถึง 100 เท่า พบว่าการสืบค้นข้อมูลโดยเฉพาะจาก 1 ตาราง และ 1 คอลเลคชัน ทั้งแบบไม่มีเงื่อนไขและแบบมีเงื่อนไข ประสิทธิภาพในการสืบค้นข้อมูลของฐานข้อมูล NoSQL เร็วกว่าฐานข้อมูลเชิงสัมพันธ์ สำหรับแนวทางในการนำฐานข้อมูล NoSQL มาใช้งานจริงในอนาคตกับระบบบริหารจัดการครุภัณฑ์ คือ ปรับแก้โครงสร้างการจัดเก็บข้อมูลจาก Referenced Document ไปเป็น Embedded Document เนื่องจาก Embedded Document จะเก็บข้อมูลที่เกี่ยวข้องทั้งหมดรวมไว้ในด็อกคิวเมนต์เดียวกันจึงค้นหาข้อมูลได้รวดเร็วกว่า และปรับแก้โค้ดเพื่อให้สามารถติดต่อและจัดการฐานข้อมูล NoSQL ได้อย่างมีประสิทธิภาพ

คำสำคัญ: ประสิทธิภาพการสืบค้นข้อมูล ฐานข้อมูลเชิงสัมพันธ์ ฐานข้อมูลโนเอสคิวแอล มายเอสคิวแอล มองโกลดีบี

ABSTRACT

This research aimed to study approaches for data storage in NoSQL databases and compare the query performance between relational databases and NoSQL databases for an information system. The case study involved the asset dataset of the Information Technology Service Center, Chiang Mai University. The relational database used in the study was MySQL, which stores data in tables consisting of rows and columns. The NoSQL database used was MongoDB, which stores data as documents with field-value pairs. PHP was used to develop a web application for testing. The researcher designed query commands and developed a web application to evaluate the query performance between the two types of databases. The initial results indicated that querying asset data using the relational database was faster than using the NoSQL database. However, the results from both databases were accurate and consistent, yielding 100% correctness. The slower performance of the NoSQL database was attributed to its data structure. When converted from a relational database, data in the NoSQL database was stored as a Referenced Document across separate collections. This required the use of functions to join data from multiple collections, leading to increased processing time and slower query performance. Further analysis was conducted to confirm that NoSQL databases could handle large-scale data more efficiently than relational databases. A larger sample database containing employee and salary data, with a data size 100 times larger than the asset dataset, was

used for additional testing. The results demonstrated that, for queries involving a single table or a single collection, both with and without conditions, NoSQL databases outperformed relational databases in query performance. To apply NoSQL databases effectively in future asset management systems, it is recommended to modify the data structure from Referenced Document to Embedded Document. This structure consolidates all related information within a single document, allowing for faster query processing. Additionally, the application code should be adjusted to support and manage NoSQL databases efficiently.

Keywords: Query Performance; Relational Database; NoSQL Database; MySQL; MongoDB

บทนำ

สำนักบริการเทคโนโลยีสารสนเทศ มหาวิทยาลัยเชียงใหม่ มีระบบสารสนเทศที่ถูกพัฒนาขึ้นสำหรับให้บริการเพื่อช่วยอำนวยความสะดวกและแบ่งเบาภาระงานประจำของเจ้าหน้าที่ส่วนงานต่างๆภายในหน่วยงานหลายระบบ เช่น ระบบแจ้งซ่อม ระบบบริหารจัดการโครงการ/กิจกรรม ระบบทะเบียนครุภัณฑ์ภายในห้องดาต้าเซ็นเตอร์ เป็นต้น ซึ่งทุกระบบใช้การบริหารจัดการข้อมูลด้วยฐานข้อมูลเชิงสัมพันธ์ทั้งสิ้น ระบบบริหารจัดการครุภัณฑ์ของสำนักบริการเทคโนโลยีสารสนเทศเป็นระบบหนึ่งซึ่งผู้วิจัยได้พัฒนาขึ้น เพื่อช่วยแบ่งเบาภาระงานของเจ้าหน้าที่งานพัสดุให้สามารถบริหารจัดการครุภัณฑ์ภายในหน่วยงานให้เป็นไปอย่างมีประสิทธิภาพ ตัวอย่างการทำงานของระบบ เช่น การเก็บบันทึกข้อมูลครุภัณฑ์ทั้งหมดของหน่วยงาน การเก็บบันทึกข้อมูลการเบิกครุภัณฑ์ของบุคลากร การเก็บบันทึกประวัติการเบิกครุภัณฑ์ของบุคลากร การสืบค้นครุภัณฑ์เพื่อตรวจนับและการแสดงรายงานครุภัณฑ์ต่างๆ เป็นต้น ซึ่งใช้ฐานข้อมูลเชิงสัมพันธ์ในการบริหารจัดการข้อมูลเช่นกัน ในการใช้งานระบบบริหารจัดการครุภัณฑ์พบว่าในทุกๆ ปีจำนวนข้อมูลครุภัณฑ์ที่มีการจัดซื้อเพิ่มเติม จำนวนข้อมูลการเบิกจ่ายครุภัณฑ์ รวมถึงจำนวนข้อมูลประวัติการเคลื่อนย้ายครุภัณฑ์จะถูกเก็บบันทึกในฐานข้อมูลในปริมาณที่เพิ่มมากขึ้นเรื่อยๆ ทำให้ขนาดของฐานข้อมูลมีขนาดใหญ่ขึ้น จำนวนข้อมูลที่เพิ่มมากขึ้นเป็นสาเหตุให้การสืบค้นหรือเข้าถึงข้อมูลทำได้ช้าลงและใช้เวลาในการประมวลผลมากขึ้น

จากปัญหาดังกล่าวผู้วิจัยจึงได้ศึกษางานวิจัยที่เกี่ยวข้องกับประสิทธิภาพในการสืบค้นข้อมูลของฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูล NoSQL ดังนี้ ชยาพร แก่นสาร [1] ได้ทำการวิเคราะห์แผนการสืบค้นเพื่อประเมินประสิทธิภาพการทำงานของออปติไมเซอร์ (เป็นกลไกในระบบจัดการฐานข้อมูลที่ใช้ในการวิเคราะห์คำสั่ง SQL [2] ซึ่งเป็นคำสั่งที่ใช้ในการอ่านเขียนข้อมูลจากฐานข้อมูลเชิงสัมพันธ์) ที่มีต่อคำสั่ง SQL แบบ Selection (คำสั่งที่ใช้ในการค้นหาข้อมูล) เพื่อประเมินประสิทธิภาพการทำงานของดีบีเอ็มเอส (ระบบซอฟต์แวร์ที่ใช้สำหรับการจัดการฐานข้อมูล) โดยใช้แผนการสืบค้น (Query Plan) และแบ่งการทำงานเป็น 4 ขั้นตอนคือ 1) จัดเตรียมข้อมูลสำหรับทดสอบและเครื่องมือที่ใช้ทดสอบได้แก่ Oracle และ SQL Developer 2) ออกแบบคำสั่ง SQL 3) ประมวลผลคำสั่ง SQL และสร้างแผนการสืบค้นและ 4) วิเคราะห์และสรุปผลการทำงาน โดยผลจากการวิเคราะห์และทดสอบพบว่ารูปแบบการเข้าถึงข้อมูล วิธีการเชื่อมโยงตาราง การใช้ฟังก์ชันภายในคำสั่ง การใช้ประโยคสืบค้นย่อยแบบวนซ้ำ จะส่งผลต่อประสิทธิภาพการเข้าถึงข้อมูล Oracle รชต ทิมสารวิชกิจ [3] ได้ทำการเปรียบเทียบการสืบค้นข้อมูลบนเทคโนโลยีข้อมูลขนาดใหญ่ที่

เชื่อว่า Hadoop กับระบบฐานข้อมูลเชิงสัมพันธ์ MySQL โดยใช้ตัวอย่างชุดข้อมูลบริการสุขภาพเป็นกรณีศึกษา เพื่อศึกษาแนวทางที่เหมาะสมในการจัดเก็บและวิเคราะห์ข้อมูลการให้บริการสุขภาพซึ่งรวบรวมโดยกระทรวงสาธารณสุขด้วยเทคนิควิธีการประมวลผลแบบแมพรีดิวและภาษา SQL (แมพรีดิวเป็นเครื่องมือที่ช่วยในประมวลผลและวิเคราะห์ข้อมูลขนาดใหญ่ในระบบที่มีการกระจายตัวของข้อมูลสูง) ทดสอบด้วยชุดข้อมูลบริการสุขภาพที่มีขนาดกระเปาะเพิ่มขึ้นเป็นลำดับ และสร้างชุดแบบสอบถามจากรายงานสรุปการเจ็บป่วยจำนวน 2 รายงาน เป็นเครื่องมือที่นำมาใช้หาประสิทธิภาพของเวลาในการประมวลผลการสืบค้นข้อมูลผลการวิจัยพบว่าเวลาในการสืบค้นข้อมูลเมื่อมีการเปรียบเทียบระหว่างเทคโนโลยีข้อมูลขนาดใหญ่กับระบบฐานข้อมูลเชิงสัมพันธ์ผลของเวลาไม่มีความแตกต่างกันทางสถิติ และผลลัพธ์จากการสืบค้นข้อมูลของเทคโนโลยีทั้งสองรูปแบบให้ผลถูกต้องแม่นยำตรงกัน 100% อานันท์ เงินบำรุง [4] ได้ทำการเปรียบเทียบประสิทธิภาพระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูล NoSQL ที่นำมาใช้เพื่อจัดเก็บข้อมูล โดยทำการวิเคราะห์ห่ออกแบบฐานข้อมูลทดสอบความเร็วในการคิวรีข้อมูล (การคิวรีข้อมูล คือการอ่านข้อมูล) ทดสอบเวลาในการนำเข้าข้อมูลและตรวจสอบการใช้ทรัพยากรของเครื่อง ผลการวิจัยพบว่าการนำเข้าข้อมูลไปยังฐานข้อมูลทั้งสองประเภทสามารถใช้เครื่องมือสำหรับการนำเข้าข้อมูลทำให้การนำเข้าข้อมูลทำได้ง่ายและรวดเร็ว การใช้ทรัพยากรของเครื่องพบว่าฐานข้อมูลเชิงสัมพันธ์มีการใช้ CPU น้อยกว่าฐานข้อมูล NoSQL แต่ใช้พื้นที่ในการเก็บข้อมูลมากกว่าฐานข้อมูล NoSQL การทดสอบความเร็วในการคิวรีข้อมูลพบว่าฐานข้อมูลเชิงสัมพันธ์สามารถคิวรีข้อมูลที่มีการเชื่อมโยงตารางมากกว่าหนึ่งตารางใช้เวลาในการคิวรีน้อยกว่าฐานข้อมูล NoSQL แต่หากเป็นการคิวรีข้อมูลจากตารางเดียวที่มีข้อมูลปริมาณมากๆ และมีการระบุข้อความเฉพาะเจาะจงสำหรับข้อมูลที่ต้องการค้นหา พบว่าฐานข้อมูล NoSQL ใช้เวลาในการคิวรีน้อยกว่าฐานข้อมูลเชิงสัมพันธ์ นรุตม์ บุตรพลอย และ พรหมเมศ วีระพันธ์ [5] ได้ทำการเปรียบเทียบระบบการจัดการฐานข้อมูลเชิงสัมพันธ์และ NoSQL กรณีศึกษาฐานข้อมูลนักศึกษามหาวิทยาลัยราชภัฏกำแพงเพชร โดยได้ทดลองเพื่อเปรียบเทียบเวลาในการสืบค้นข้อมูลที่ซับซ้อน และความยืดหยุ่นในการปรับเปลี่ยนโครงสร้างของฐานข้อมูล ระหว่างฐานข้อมูลเชิงสัมพันธ์ คือ MySQL กับฐานข้อมูล NoSQL คือ MongoDB โดยข้อมูลที่ใช้ในการทดสอบสำหรับฐานข้อมูล NoSQL เป็นข้อมูลนักศึกษาที่มีโครงสร้างเป็นแบบ Embedded Document ผลการทดลองพบว่า MongoDB ใช้เวลาในการสืบค้นข้อมูลที่เร็วกว่าและการปรับเปลี่ยนโครงสร้างมีความยืดหยุ่นมากกว่าเมื่อเปรียบเทียบกับ MySQL แต่ในการศึกษาพบว่าฐานข้อมูล NoSQL ไม่ใช่ทางเลือกที่จะนำมาแทนที่ระบบฐานข้อมูลเชิงสัมพันธ์ของกรณีศึกษา เนื่องจากการศึกษารายละเอียดของระบบงานพบว่าข้อมูลเพิ่มขึ้นเป็นช่วงเวลาและไม่ได้เกิดขึ้นทุกวัน ปริมาณข้อมูลไม่ได้มีจำนวนมากจนต้องเปลี่ยนมาใช้ฐานข้อมูล NoSQL

นอกจากนี้ผู้วิจัยพบว่าในปัจจุบันความก้าวหน้าทางด้านเทคโนโลยีสำหรับจัดเก็บข้อมูลที่เรียกว่าฐานข้อมูล NoSQL ได้ถูกพัฒนาขึ้นเพื่อให้สามารถจัดการกับข้อมูลที่มีขนาดใหญ่ การจัดเก็บข้อมูลมีความเป็นอิสระ ไม่จำเป็นต้องอาศัยการกำหนดรูปแบบหรือประเภทของข้อมูล เป็นฐานข้อมูลที่ใช้ทรัพยากรของเครื่องน้อย สามารถประมวลผลได้อย่างรวดเร็วและช่วยลดพลังงานในการประมวลผล จากความสามารถในการประมวลผลและการสืบค้นของฐานข้อมูล NoSQL ที่อาจนำมาใช้เพื่อช่วยแก้ปัญหาข้อจำกัดที่พบในการใช้ฐานข้อมูลเชิงสัมพันธ์ ผู้วิจัยมีความสนใจที่จะศึกษาแนวทางในการจัดเก็บข้อมูล NoSQL รวมถึงเปรียบเทียบประสิทธิภาพการสืบค้นข้อมูลระหว่างฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูล NoSQL ของระบบบริหารจัดการครุภัณฑ์เพื่อใช้เป็นข้อมูลศึกษา ซึ่งผลที่ได้จากการศึกษาจะสามารถแปลงข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปเป็นฐานข้อมูล NoSQL และนำข้อมูลที่อยู่ในฐานข้อมูล NoSQL มาปรับใช้งานร่วมกับระบบ

บริหารจัดการครุภัณฑ์ที่จะพัฒนา ปรับปรุง ขยายการให้บริการแก่คณะหน่วยงานต่างๆภายในมหาวิทยาลัย ในอนาคต โดยคาดว่าปริมาณข้อมูลครุภัณฑ์ที่ถูกจัดเก็บในฐานข้อมูลเชิงสัมพันธ์จะเพิ่มมากขึ้นเรื่อยๆ นอกจากนี้ยังสามารถใช้เป็นแนวทางในการแก้ปัญหาสำหรับการนำฐานข้อมูล NoSQL มาใช้ร่วมกับระบบสารสนเทศอื่นๆ เช่น ระบบทะเบียนครุภัณฑ์ภายในห้องดาต้าเซ็นเตอร์ แทนการใช้ฐานข้อมูลเชิงสัมพันธ์ เพื่อช่วยเพิ่มประสิทธิภาพในเข้าถึงและการบริหารจัดการข้อมูลและเพิ่มประสิทธิภาพในการสืบค้นข้อมูล

ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

1. ฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ [6] หรือ Relational Database เป็นรูปแบบการจัดเก็บข้อมูลที่รวบรวมข้อมูลที่มีความสัมพันธ์กันไว้ด้วยกันในรูปแบบของตาราง 2 มิติ ฐานข้อมูลหนึ่งอาจประกอบด้วยตารางหลายๆ ตารางและในแต่ละตารางจะมีความสัมพันธ์ (Relation) ซึ่งกันและกัน โดยในแต่ละตารางจะประกอบด้วยแถวและคอลัมน์

ฐานข้อมูลเชิงสัมพันธ์ประกอบด้วยองค์ประกอบ 3 ส่วน ดังนี้

- 1) ส่วนโครงสร้างของข้อมูลเป็นส่วนของการจัดเก็บข้อมูลในรูปแบบของตารางที่มีความสัมพันธ์กัน โดยจัดเก็บเป็นแถวและคอลัมน์
- 2) ส่วนควบคุมความคงสภาพของข้อมูลเป็นส่วนที่สร้างขึ้นมาเพื่อใช้ในการกำหนดความสัมพันธ์ของข้อมูลให้มีความคงสภาพ ได้แก่ การกำหนดคีย์หลัก (Primary Key) และคีย์นอก (Foreign Key)
- 3) ส่วนจัดการข้อมูลเป็นส่วนของคำสั่งที่ใช้จัดการกับข้อมูลที่เก็บบันทึกอยู่ในฐานข้อมูลโดยมีภาษาที่ใช้ในการจัดการกับฐานข้อมูลคือ SQL (Structured Query Language)

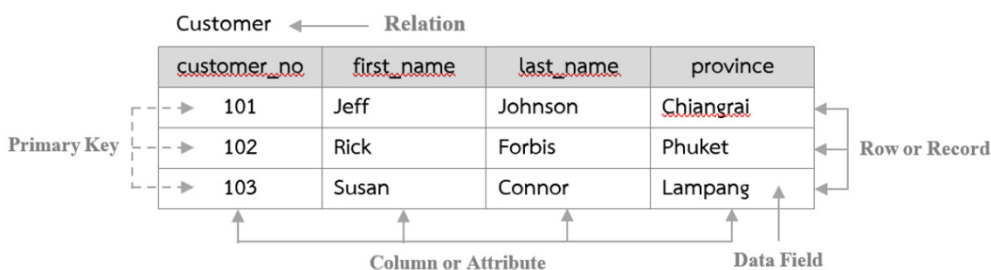


Figure 1 Example of the relationship of the Customer table in a relational database

2. ฐานข้อมูล NoSQL

ฐานข้อมูล NoSQL [7] (Not only SQL) หรือ Non-Relational Database หรือฐานข้อมูลที่ไม่ใช่เชิงสัมพันธ์ เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นสำหรับเก็บข้อมูลจำนวนมาก (Big Data) มีรูปแบบการจัดเก็บข้อมูลซึ่งต่างจากฐานข้อมูลเชิงสัมพันธ์ ไม่มีโครงสร้างข้อมูลที่ตายตัว มีความยืดหยุ่นในการจัดการข้อมูล สามารถประมวลผลข้อมูลที่มีขนาดใหญ่ได้อย่างรวดเร็วและมีประสิทธิภาพ ปัจจุบันได้ถูกนำมาใช้กับแอปพลิเคชันต่างๆ แทนที่ฐานข้อมูลเชิงสัมพันธ์มากขึ้น เพื่อแก้ปัญหาและข้อจำกัดที่ไม่สามารถแก้ไขด้วยการใช้ฐานข้อมูลเชิงสัมพันธ์ ตัวอย่างเช่น

- การรองรับ Big Data ฐานข้อมูลเชิงสัมพันธ์อาจมีข้อจำกัดในการจัดการกับข้อมูลที่มีขนาดใหญ่เพิ่มขึ้นเรื่อยๆ ในขณะที่ฐานข้อมูล NoSQL ถูกออกแบบมาเพื่อรองรับ Big Data โดยเฉพาะ

- ความเร็วในการเข้าถึงข้อมูลจำนวนมาก ฐานข้อมูลเชิงสัมพันธ์ใช้คำสั่ง SQL ในการเข้าถึงข้อมูลซึ่งอาจซับซ้อนและช้าหากข้อมูลมีขนาดใหญ่หรือไม่มีการสร้างดัชนีอย่างเหมาะสม ในขณะที่ฐานข้อมูล NoSQL จะเข้าถึงข้อมูลแบบเรียบง่าย เช่น Key-Value ซึ่งเร็วกว่าในกรณีที่ไม่มีความสัมพันธ์ระหว่างข้อมูล

- ความสามารถในการขยายตัว ฐานข้อมูลเชิงสัมพันธ์มักขยายโดยการเพิ่มทรัพยากรให้กับเซิร์ฟเวอร์เดียว ซึ่งมีข้อจำกัดเมื่อข้อมูลมีปริมาณเพิ่มมากขึ้น ในขณะที่ฐานข้อมูล NoSQL รองรับการขยายแบบกระจาย โดยสามารถเพิ่มเซิร์ฟเวอร์เพื่อกระจายการจัดเก็บและการประมวลผล

การวิจัยนี้ใช้ฐานข้อมูล NoSQL ประเภทฐานข้อมูลเอกสาร (Document Database) ที่มีการจัดเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) โดยโครงสร้างการเก็บข้อมูลจะเก็บข้อมูลในรูปแบบของคีย์-ค่า (Key-Value) ฐานข้อมูล NoSQL ประเภทนี้ได้แก่ MongoDB CouchDB เป็นต้น

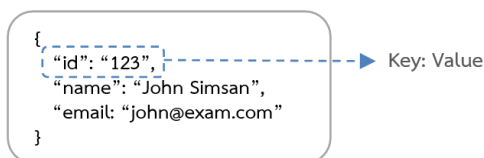


Figure 2 JSON data sample

3. MongoDB

MongoDB [7, 8] เป็นฐานข้อมูล NoSQL ประเภทฐานข้อมูลเอกสารเป็นฐานข้อมูลที่ไม่มีความสัมพันธ์เชิงโครงสร้าง ต่างจากฐานข้อมูลเชิงสัมพันธ์ และเป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อให้สามารถจัดการกับข้อมูลที่มีขนาดใหญ่ เป็นฐานข้อมูลที่ใช้ทรัพยากรของเครื่องน้อย การประมวลผลทำได้รวดเร็วและมีความยืดหยุ่นในการใช้งานสามารถนำมาใช้ในการจัดเก็บข้อมูลของระบบสารสนเทศต่างๆ เช่น เว็บแอปพลิเคชันได้ โดยโครงสร้างของฐานข้อมูล MongoDB ประกอบด้วย

คอลเลคชัน (Collection) หรือกลุ่มเอกสารเป็นที่เก็บรวบรวมดีออกคิวเมนต์หลายๆ ดีออกคิวเมนต์ คล้ายกับตารางในฐานข้อมูลเชิงสัมพันธ์แต่ไม่มีโครงสร้างการจัดเก็บข้อมูลที่ตายตัว

ดีออกคิวเมนต์ (Document) หรือเอกสารเป็นข้อมูลพื้นฐานที่เก็บข้อมูลในรูปแบบ BSON (Binary JSON) รูปแบบการเข้ารหัสข้อมูลไบนารีรูปแบบหนึ่งของ JSON แต่ละดีออกคิวเมนต์ในคอลเลคชันอาจมีโครงสร้างที่แตกต่างกัน

ฟิลด์-ค่า (Field-Value) เป็นส่วนของข้อมูลพื้นฐานในดีออกคิวเมนต์ ข้อมูลจะเก็บในรูปแบบ BSON คล้ายกับคอลัมน์ในฐานข้อมูลเชิงสัมพันธ์ รูปที่ 3 แสดงโครงสร้างการเก็บข้อมูลของ MongoDB โดย MongoDB สามารถเก็บข้อมูลได้ทั้งในรูปแบบของดีออกคิวเมนต์ อาร์เรย์ หรืออาร์เรย์ของดีออกคิวเมนต์ (Array of Document)

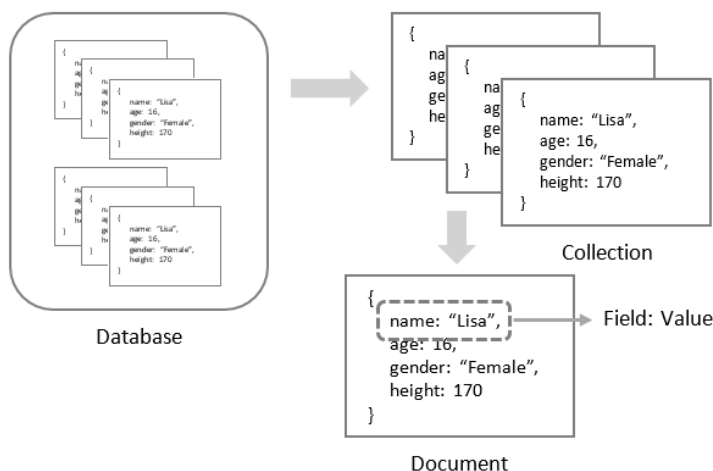


Figure 3 Data structure of MongoDB [7]

การกำหนดโครงสร้างหรือความสัมพันธ์ของด็อกคิวเมนต์ใน MongoDB นั้นสามารถจำแนกโครงสร้างด็อกคิวเมนต์ได้เป็น 2 แบบคือ

1) ด็อกคิวเมนต์แบบฝังตัว (Embedded Document) เป็นด็อกคิวเมนต์ที่เก็บข้อมูลที่เกี่ยวข้องหรือมีความสัมพันธ์กันไว้ในด็อกคิวเมนต์เดียวกัน เทียบได้กับการเก็บข้อมูลที่เกิดจากการเชื่อมโยง (Join) ตารางหลายตารางที่สัมพันธ์กันของฐานข้อมูลเชิงสัมพันธ์เข้าด้วยกัน โดย Embedded Document สามารถเก็บข้อมูลที่มีความสัมพันธ์กันในหลายรูปแบบเช่น ฟิลด์ อาร์เรย์ หรือเก็บข้อมูลแบบเอกสารซ้อนกัน (Nested Document) และสามารถเก็บข้อมูลที่มีความซ้ำซ้อนกันได้ เหมาะสำหรับข้อมูลที่ไม่เปลี่ยนแปลงบ่อย ข้อดีของการเก็บข้อมูลในรูปแบบนี้คือการเข้าถึงข้อมูล เช่น การอ่านหรือเขียนข้อมูลทำได้รวดเร็ว เนื่องจากข้อมูลต่างๆจะถูกจัดเก็บไว้ในด็อกคิวเมนต์เดียว ข้อเสียคือการแก้ไขข้อมูลทำได้ช้าเนื่องจากข้อมูลที่เก็บมีความซ้ำซ้อน การแก้ไขจึงต้องแก้ไขข้อมูลที่ซ้ำซ้อนกันหลายที่ การกำหนดความสัมพันธ์ของด็อกคิวเมนต์สามารถกำหนดได้ทั้งแบบ One-to-one หรือ One-to-many ตัวอย่าง Embedded Document แสดงดังรูปที่ 4

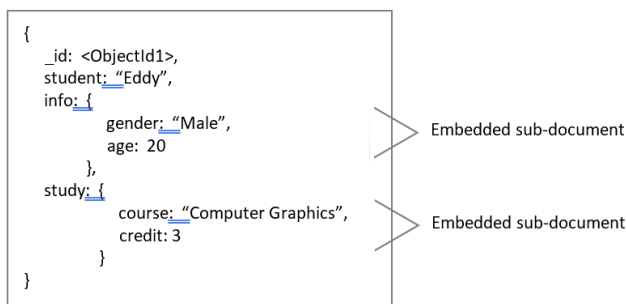


Figure 4 Structure of Embedded Document [7]

2) ด็อกคิวเมนต์แบบอ้างอิง (Referenced Document) เป็นด็อกคิวเมนต์ที่เก็บข้อมูลที่เกี่ยวข้องแยกออกจากกันเป็นคอลเลกชันอิสระและเชื่อมโยงกันด้วยคีย์อ้างอิง (Referenced Key) เช่น ID และสามารถใช้อ้างอิงที่อยู่ในด็อกคิวเมนต์หนึ่ง เพื่อเชื่อมโยงข้อมูลไปยังอีกด็อกคิวเมนต์ที่มีคีย์อ้างอิงเดียวกัน ข้อดีคือการแก้ไขข้อมูลสามารถแก้ไขเพียงที่เดียวเนื่องจากข้อมูลไม่ซ้ำกัน ข้อเสียคือการเพิ่มข้อมูลต้องเพิ่มหลายที่เพื่อให้ข้อมูลสัมพันธ์กัน หากโครงสร้างการเก็บข้อมูลมีความซับซ้อน การอ่านข้อมูลจะอ่านได้ช้าลง เนื่องจากต้องทำการเชื่อมข้อมูลจากหลายคอลเลกชัน การกำหนดความสัมพันธ์ของด็อกคิวเมนต์สามารถกำหนดได้เป็นแบบ One-to-many ตัวอย่าง Referenced Document แสดงดังรูปที่ 5

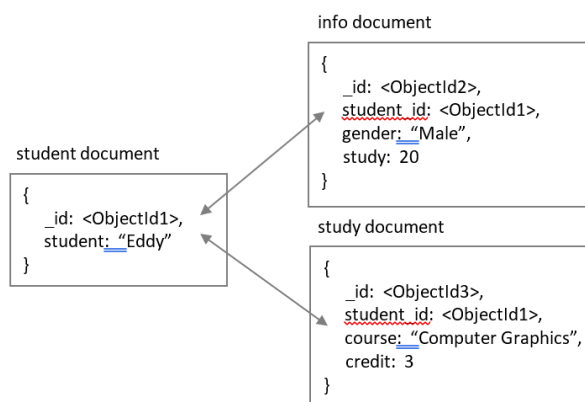


Figure 5 Structure of Referenced Document [7]

วิธีดำเนินการวิจัย

1. กรอบแนวคิด

ระบบบริหารจัดการครุภัณฑ์ของสำนักบริการเทคโนโลยีสารสนเทศมหาวิทยาลัยเชียงใหม่เป็นระบบที่ถูกพัฒนาขึ้นเพื่อช่วยแบ่งเบาภาระงานของเจ้าหน้าที่งานพัสดุให้สามารถบริหารจัดการข้อมูลครุภัณฑ์ภายในหน่วยงานให้เป็นไปอย่างมีประสิทธิภาพ ระบบมีการบริหารจัดการข้อมูลโดยใช้ฐานข้อมูลเชิงสัมพันธ์ซึ่งการใช้งานระบบพบว่าในแต่ละปีข้อมูลต่างๆ เช่น ข้อมูลครุภัณฑ์ ข้อมูลการเบิกจ่ายครุภัณฑ์ ข้อมูลประวัติการเคลื่อนย้ายครุภัณฑ์จะถูกบันทึกเก็บในฐานข้อมูลเพิ่มมากขึ้น ทำให้ฐานข้อมูลมีขนาดใหญ่ขึ้นเรื่อยๆ จำนวนข้อมูลที่มีปริมาณมากอาจส่งผลกระทบต่อการใช้งานข้อมูลและส่งผลต่อประสิทธิภาพในการสืบค้น การสืบค้นทำได้ช้าลงและใช้เวลาประมวลผลมากขึ้น ดังนั้นจึงมีการนำฐานข้อมูล NoSQL ที่มีความสามารถในการจัดการกับข้อมูลที่มีขนาดใหญ่ การจัดเก็บข้อมูลไม่จำเป็นต้องกำหนดรูปแบบหรือประเภทของข้อมูล การสืบค้นข้อมูลทำได้รวดเร็ว และเป็นฐานข้อมูลที่ใช้ทรัพยากรของเครื่องน้อย มาทดลองเพื่อปรับใช้กับระบบแทนฐานข้อมูลเชิงสัมพันธ์

2. ขั้นตอนการดำเนินงาน

2.1 การจัดเตรียมเครื่องมือที่ใช้ในการวิจัยประกอบด้วย ซอฟต์แวร์ และฮาร์ดแวร์ ดังต่อไปนี้

1) MySQL [9] เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ซึ่งใช้ภาษา SQL ในการจัดการกับข้อมูลและถูกนำมาใช้ในการทดสอบประสิทธิภาพการสืบค้นข้อมูลเวอร์ชันที่ใช้ คือ 14.14 Distrib 5.7.17

2) Navicat [10] เป็นโปรแกรมที่ใช้สำหรับบริหารจัดการฐานข้อมูลเชิงสัมพันธ์ ในที่นี้คือ MySQL มีลักษณะการใช้งานเป็นกราฟิกติดต่อกับการทำงานกับผู้ใช้ มีเครื่องมือต่างๆ ที่ง่ายต่อการจัดการกับข้อมูล โปรแกรมนี้เป็นโปรแกรมที่ผู้วิจัยใช้สำหรับจัดการข้อมูลที่อยู่ในฐานข้อมูล MySQL ในการพัฒนาระบบสารสนเทศต่างๆ ภายในหน่วยงาน รวมถึงระบบบริหารจัดการครุภัณฑ์โดยจะใช้โปรแกรมนี้ในการคิวรีคำสั่งเพื่อทดสอบประสิทธิภาพการสืบค้นข้อมูลของ MySQL เวอร์ชันที่ใช้ คือ 8.2.19

3) MongoDB เป็นระบบจัดการฐานข้อมูล NoSQL ประเภทฐานข้อมูลเอกสารซึ่งใช้ภาษา MongoDB Query Language [11] ในการจัดการกับข้อมูลและถูกนำมาใช้ในการทดสอบประสิทธิภาพการสืบค้นข้อมูล เวอร์ชันที่ใช้ คือ 4.0.11

4) Studio 3T [12] เป็นโปรแกรมที่ใช้สำหรับบริหารจัดการฐานข้อมูล NoSQL ในที่นี้คือ MongoDB มีลักษณะการใช้งานเป็นกราฟิกติดต่อกับการทำงานกับผู้ใช้ มีเครื่องมือต่างๆ ที่ง่ายต่อการจัดการกับข้อมูล และมีเครื่องมือที่ง่ายต่อการแปลงข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ให้เป็นฐานข้อมูล NoSQL โดยจะใช้โปรแกรมนี้ในการแปลงข้อมูลครุภัณฑ์จากฐานข้อมูล MySQL มาเก็บในฐานข้อมูล MongoDB และคิวรีคำสั่งเพื่อทดสอบประสิทธิภาพการสืบค้นข้อมูลของ MongoDB เวอร์ชันที่ใช้ คือ 2022.2.0

5) PHP เป็นภาษาที่ใช้พัฒนาเว็บแอปพลิเคชันเพื่อทดสอบเปรียบเทียบประสิทธิภาพการสืบค้นข้อมูลครุภัณฑ์ระหว่างฐานข้อมูลเชิงสัมพันธ์ [13] กับฐานข้อมูล NoSQL [14] เวอร์ชันที่ใช้ คือ 7.4.27

6) เครื่องคอมพิวเตอร์โน้ตบุ๊กสำหรับใช้จำลองเป็น Web Application Server และติดตั้งระบบที่ใช้ในการทดสอบจำนวน 1 เครื่องโดยมีรายละเอียดดังนี้

- หน่วยประมวลผลกลาง: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx 2.00 GHz
- หน่วยความจำหลัก: 8 GB (7.55 GB usable)
- ระบบปฏิบัติการ: Windows 10
- Web Application Server: Internet Information Services (IIS) เวอร์ชัน 10.0.19041.1

2.2 การจัดเตรียมข้อมูลและแปลงข้อมูลเพื่อใช้ในการทดสอบ

การจัดเตรียมข้อมูลที่ใช้ในการทดสอบจะเตรียมฐานข้อมูล 2 ชุดดังนี้

ชุดที่ 1 เป็นข้อมูลครุภัณฑ์ที่เก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์ซึ่งใช้ระบบจัดการฐานข้อมูล MySQL ในการเก็บบันทึกและจัดการกับข้อมูล และฐานข้อมูลที่ใช้ในการทดสอบมีชื่อว่า asset ประกอบด้วยตารางหลายตารางซึ่งเก็บข้อมูลครุภัณฑ์ต่างๆที่มีความสัมพันธ์กันไว้ด้วยกันแสดงดังรูปที่ 6 และตัวอย่างข้อมูลครุภัณฑ์ที่เก็บในตารางแสดงดังรูปที่ 7

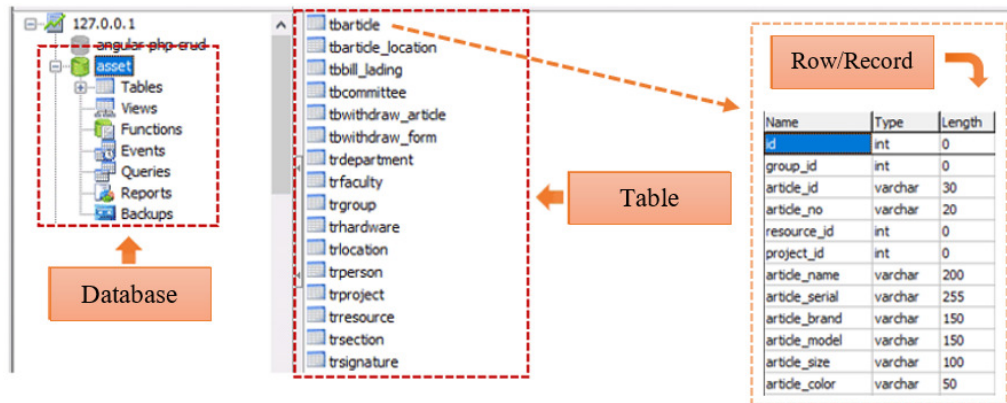


Figure 6 Relational database MySQL named asset used for testing

| id | group_id | article_id | article_no | resource_id | project_id | article_name | article_serial | article_brand | article_model |
|----|----------|------------------------|------------|-------------|------------|---|----------------|---------------|---------------|
| 87 | 7 | 51.74400010001.0251028 | 1132065 | 2 | 0 | คอมพิวเตอร์ HP dc5700 small Form Factor | | | |
| 88 | 1 | 51.71100040003.0247001 | 1132071 | 2 | 0 | ตู้เซฟ | | | |
| 89 | 8 | 51.67200050006.0248001 | 1132108 | 2 | 0 | กล้องถ่ายภาพวิดีโอ | | | |
| 90 | 7 | 51.74400100001.0249001 | 1132113 | 2 | 0 | HP Laserjet 2605 dn Printer | | | |
| 91 | 7 | 51.74400100001.0250001 | 1132146 | 2 | 0 | Printer HP 2600n Color Laserjet | | | |
| 92 | 7 | 51.74400010001.0251002 | 1132168 | 2 | 0 | Printer hp laserjet m3035xs mfp | | | |
| 93 | 7 | 51.74400010001.0251001 | 1132171 | 2 | 0 | Scanjet HP 5590 | | | |

Figure 7 Example of asset data stored in the relational database table of MySQL

ชุดที่ 2 เป็นข้อมูลครุภัณฑ์ที่ถูกแปลงจากฐานข้อมูลเชิงสัมพันธ์ MySQL ด้วยโปรแกรม Studio 3T ให้อยู่ในรูปแบบของฐานข้อมูล NoSQL ซึ่งใช้ระบบจัดการฐานข้อมูล MongoDB ในการเก็บบันทึกและจัดการกับข้อมูล และฐานข้อมูลที่ใช้ในการทดสอบมีชื่อว่า asset_mongo ประกอบด้วยคอลเลคชันต่างๆ เป็นอิสระแยกจากกัน และเชื่อมโยงความสัมพันธ์กันด้วยคีย์อ้างอิง โดยคีย์อ้างอิงนี้ก็คือ Primary Key หรือ Foreign Key ในตารางของฐานข้อมูลเชิงสัมพันธ์นั่นเอง แสดงดังรูปที่ 8 และตัวอย่างข้อมูลครุภัณฑ์ที่เก็บในคอลเลคชันแสดงดังรูปที่ 9

หมายเหตุ: ทุกคีย์คิควเมนทจะมี_id ซึ่งเป็นค่าที่ถูกสร้างขึ้นโดยอัตโนมัติและจะมีค่าไม่ซ้ำกัน

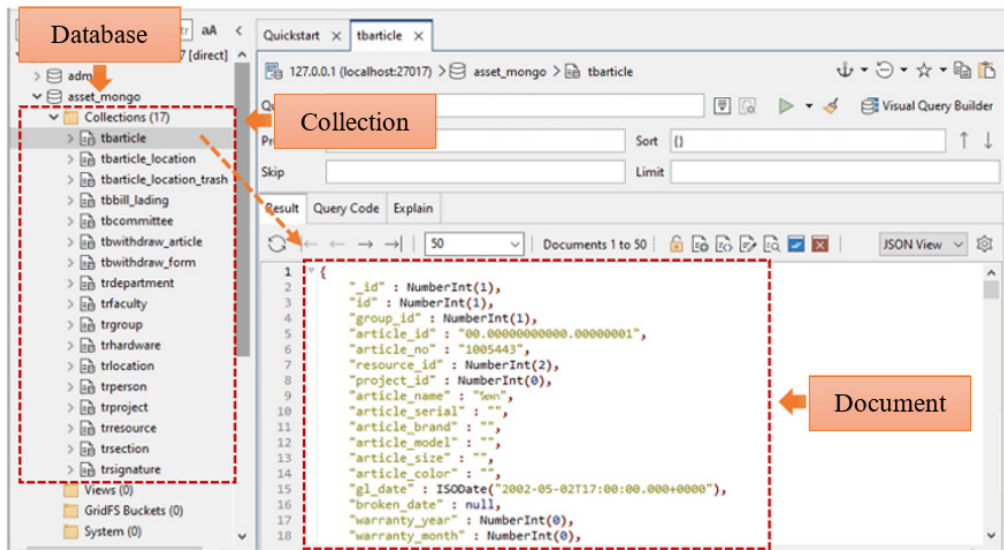


Figure 8 NoSQL MongoDB database named asset_mongo used for testing

```

{
  "_id" : NumberInt(1),
  "id" : NumberInt(1),
  "group_id" : NumberInt(1),
  "article_id" : "00.0000000000.0000000",
  "article_no" : "1005443",
  "resource_id" : NumberInt(2),
  "project_id" : NumberInt(0),
  "article_name" : "โหม",
  "article_serial" : "",
  .....
}

{
  "_id" : NumberInt(2),
  "id" : NumberInt(2),
  "group_id" : NumberInt(6),
  "article_id" : "00.0000000000.0000000",
  "article_no" : "1005445",
  "resource_id" : NumberInt(2),
  "project_id" : NumberInt(0),
  "article_name" : "ชุดเอนิเตอร์",
  "article_serial" : "",
  .....
}

{
  "_id" : NumberInt(3),
  "id" : NumberInt(3),
  "group_id" : NumberInt(7),
  "article_id" : "00.0000000000.0000000",
  "article_no" : "1005446",
  "resource_id" : NumberInt(2),
  "project_id" : NumberInt(0),
  "article_name" : "เครื่องพิมพ์เลเซอร์",
  "article_serial" : "",
  .....
}
    
```

Figure 9 Example of asset data stored in the NoSQL collection of MongoDB

โดยการแปลงข้อมูลเพื่อให้ได้ฐานข้อมูลชุดที่ 2 ด้วยโปรแกรม Studio 3T จะใช้วิธีการ Import ข้อมูลจากฐานข้อมูล MySQL และแปลงให้เป็นข้อมูล BSON เก็บลงในฐานข้อมูล MongoDB หรือเรียกว่าเป็นการทำ SQL Migration คือ เป็นการย้ายข้อมูลจากฐานข้อมูลระบบหนึ่งไปยังฐานข้อมูลอีกระบบหนึ่ง ซึ่งข้อมูลที่ถูกลำมายังฐานข้อมูล MongoDB จะมีจำนวนคอลเลคชันเท่ากับจำนวนตาราง มีจำนวนด็อกคิวเมนต์เท่ากับจำนวนแถวในฐานข้อมูล MySQL รวมถึงมีชื่อตารางชื่อคอลเลคชันและมีข้อมูลที่เหมือนกัน โดยมีขั้นตอนการแปลงข้อมูลดังนี้

- 1) เปิดใช้งานโปรแกรม Studio 3T คลิกที่ Connect จะปรากฏไดอะล็อก Connection Manager ให้คลิกที่ localhost:27017 เพื่อเชื่อมต่อฐานข้อมูล MongoDB ผ่าน Port 27017 ดังรูปที่ 10

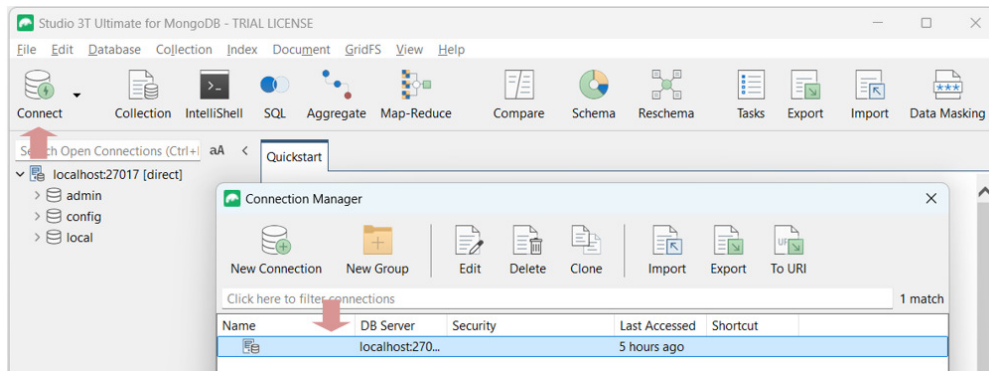


Figure 10 Connecting to use MongoDB database in Studio 3T

2) สร้างฐานข้อมูล MongoDB ที่ใช้ในการทดสอบ โดยคลิกขวาที่ localhost:27017 จะปรากฏเมนูให้คลิกที่ Add Database ดังรูปที่ 11 และตั้งชื่อฐานข้อมูลว่า asset_mongo ดังรูปที่ 12



Figure 11 Creating MongoDB database use for testing

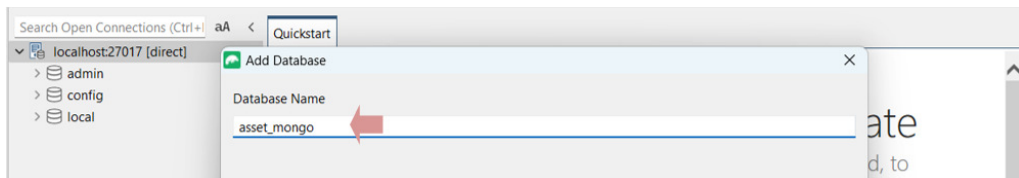


Figure 12 Set the name of the MongoDB database to asset_mongo

3) คลิกขวาที่ฐานข้อมูลชื่อ asset_mongo จะปรากฏเมนูให้คลิกที่ SQL Migration และ SQL -> MongoDB Migration เพื่อทำการ Import ข้อมูลจาก MySQL ไปยัง MongoDB ดังรูปที่ 13

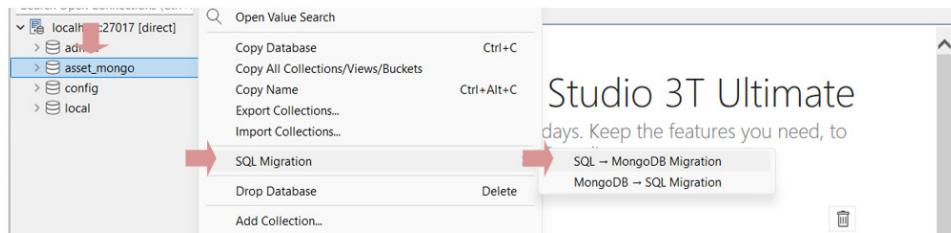


Figure 13 Importing data from MySQL to MongoDB

4) คลิกแท็บ Overview และคลิกที่ลิงค์ Click here to connect to a server... จะปรากฏไดอะล็อก SQL Connection Manager ให้ทำการสร้างการเชื่อมต่อไปยังฐานข้อมูล MySQL โดยคลิกที่ New Connection ดังรูปที่ 14

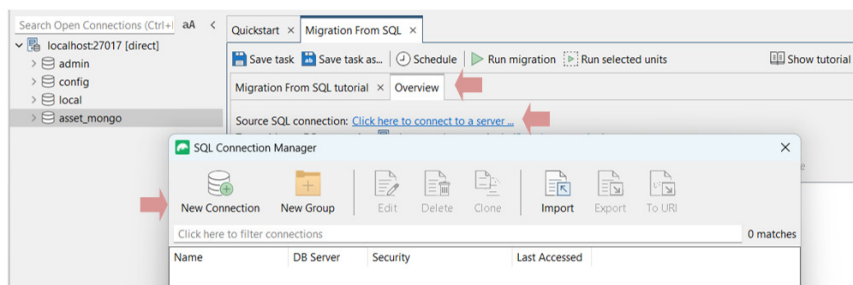


Figure 14 Creating the new connection to MySQL for data migration

5) จะปรากฏไดอะล็อก New SQL Connection ให้พิมพ์ข้อมูล Host, Username, Password และระบุ Database จาก MySQL ที่ต้องการ Import ข้อมูลมายัง MongoDB ผ่าน Port 3306 ในที่นี้ฐานข้อมูลที่จะ Import ชื่อ asset แล้วคลิกที่ปุ่ม Save ดังรูปที่ 15 (A) จะปรากฏไดอะล็อก SQL Connection Manager ให้คลิกเลือกที่ 127.0.0.1:3306 เพื่อเชื่อมต่อไปยังฐานข้อมูล MySQL แล้วคลิกที่ปุ่ม OK ดังรูปที่ 15 (B)

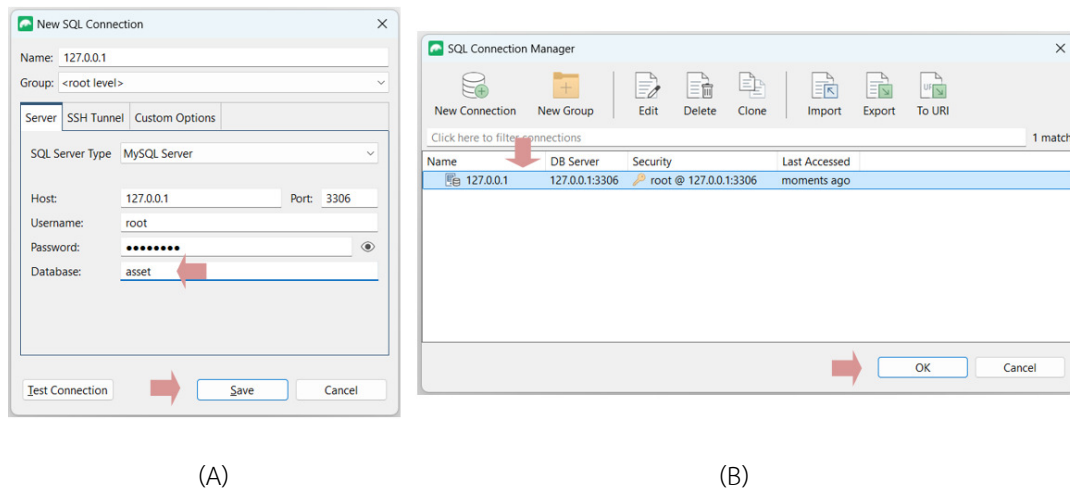


Figure 15 (A) Saving the database to import data (B) Connecting to MySQL

6) เมื่อทำการเชื่อมต่อฐานข้อมูล MySQL ได้สำเร็จ คลิกที่ Add from source table ดังรูปที่ 16

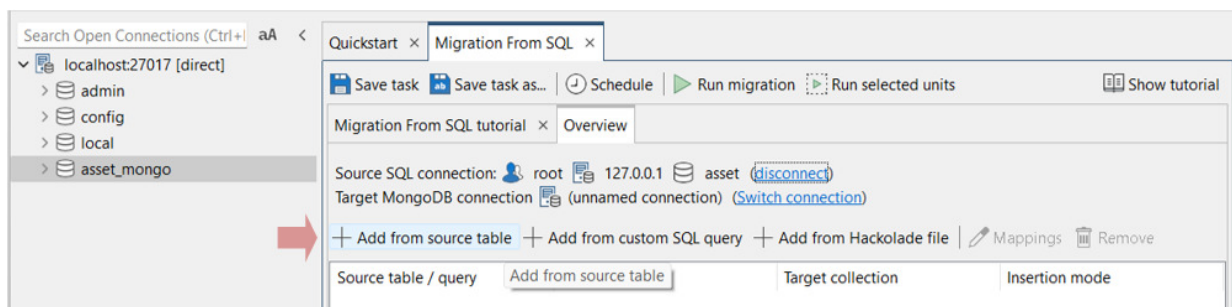


Figure 16 Adding the tables to import data

7) จะปรากฏไดอะล็อก Select table to import ให้คลิกเลือกตารางที่ต้องการ Import ในที่นี้จะเลือกตารางทั้งหมดที่อยู่ในฐานข้อมูลชื่อ asset แล้วคลิกที่ปุ่ม OK ดังรูปที่ 17

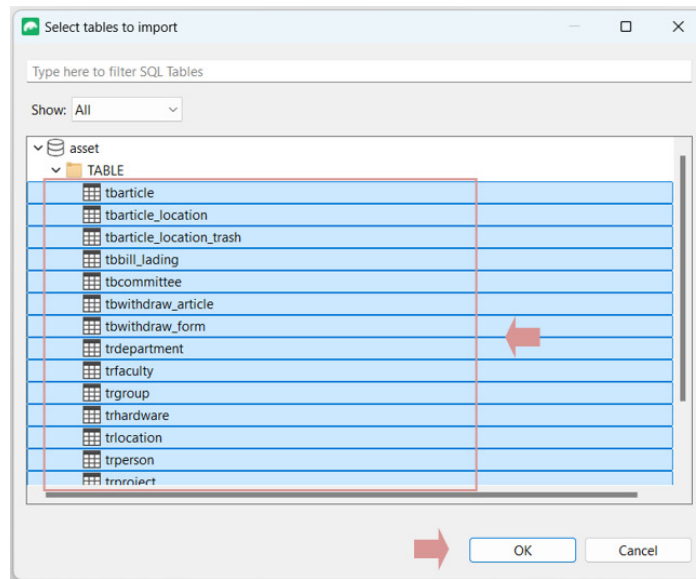


Figure 17 Select all tables from asset database to import into asset_mongo database

8) โปรแกรม Studio 3T จะแสดงตารางทั้งหมดที่ถูกเลือกจากฐานข้อมูลชื่อ asset ของ MySQL จากนั้นคลิกที่ Run migration เพื่อย้ายหรือ Import ข้อมูลดังรูปที่ 18

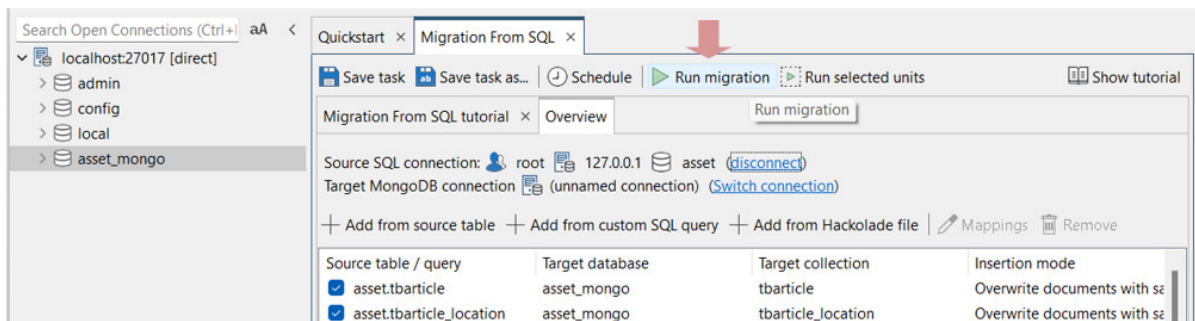


Figure 18 Importing data from MySQL database to MongoDB database

9) โปรแกรม Studio 3T จะทำการ Import ตารางและข้อมูลต่างๆ ทั้งหมดจากฐานข้อมูล asset มาไว้ในฐานข้อมูล asset_mongo ดังรูปที่ 19

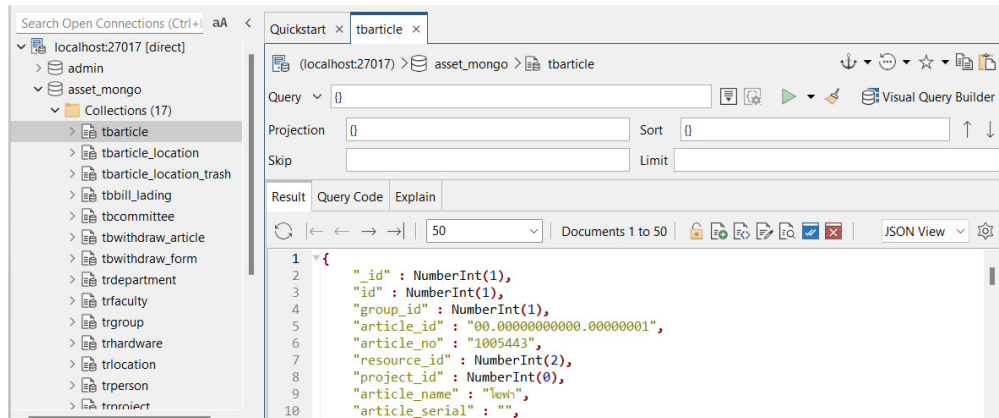


Figure 19 Database and collections imported from asset database into asset_mongo database

จะเห็นว่าข้อมูลชุดที่ 1 เป็นข้อมูลครุภัณฑ์ที่เก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์ซึ่งการออกแบบตารางแต่ตารางจะมีการสร้าง Primary Key หรือ Foreign Key เพื่อใช้เชื่อมโยงความสัมพันธ์ระหว่างตาราง เช่น article_id เป็น Primary Key ของตาราง tbarcode ดังนั้นเมื่อข้อมูลชุดที่ 1 ถูกแปลงมาเป็นข้อมูลชุดที่ 2 และเก็บอยู่ในฐานข้อมูล NoSQL คอลเลคชันที่ถูกแปลงมาจึงประกอบไปด้วยคีย์อ้างอิงโดยอัตโนมัติ เช่น article_id เป็นคีย์อ้างอิงในคอลเลคชัน tbarcode เพื่อใช้เชื่อมโยงความสัมพันธ์ระหว่างคอลเลคชัน โดยข้อมูลชุดที่ 2 นี้จะถูกแปลงให้อยู่ในรูปแบบของ Referenced Document

2.3 การออกแบบคำสั่งเพื่อทดสอบประสิทธิภาพการสืบค้น

ผู้วิจัยได้เลือกตารางและคอลเลคชันที่นำมาใช้ในการทดสอบได้แก่ ข้อมูลครุภัณฑ์ ข้อมูลการเบิกครุภัณฑ์ ข้อมูลผู้เบิก/ผู้ดูแลครุภัณฑ์ และข้อมูลชื่อสถานที่ตั้งครุภัณฑ์ตามลำดับ แสดงดังตารางที่ 1

Table 1 Displays the tables and collections used for testing of MySQL and MongoDB databases respectively

| MySQL Database (asset) | | | MongoDB Database (asset_mongo) | | |
|------------------------|------------------|--------|--------------------------------|------------------|-----------|
| Table | Data stored | Rows | Collection | Data stored | Documents |
| tbarcode | Asset | 6,025 | tbarcode | Asset | 6,025 |
| tbarcode_location | Asset withdrawal | 17,066 | tbarcode_location | Asset withdrawal | 17,066 |
| trperson | Asset owner | 125 | trperson | Asset owner | 125 |
| trlocation | Asset location | 97 | trlocation | Asset location | 97 |

ทำการออกแบบคำสั่งเพื่อใช้วัดความเร็วในการสืบค้นข้อมูลจากฐานข้อมูล MySQL ด้วยภาษา SQL และออกแบบคำสั่งเพื่อใช้วัดความเร็วในการสืบค้นข้อมูลจากฐานข้อมูล MongoDB ด้วยภาษา MongoDB Query Language ตัวอย่างคำสั่งแสดงในตารางที่ 2 ในกรณีของฐานข้อมูล MongoDB เนื่องจากข้อมูลครุภัณฑ์ที่นำมาทดสอบทั้ง 4 คอลเลคชัน เป็นข้อมูลที่ได้มาจากการแปลงข้อมูลจากตารางในฐานข้อมูล MySQL โดย

เป็นการแปลงข้อมูลจาก 1 ตารางให้เป็น 1 คอลเลคชัน และข้อมูลที่ได้มีโครงสร้างต่อกิวเมนต์เป็นแบบ Referenced Document ซึ่งต่อกิวเมนต์ที่เกี่ยวข้องกันจะถูกแยกเป็นอิสระจากกัน ดังนั้นในกรณีทดสอบที่ต้องมีการเชื่อมข้อมูลระหว่างคอลเลคชันจึงได้ใช้ฟังก์ชัน aggregate เพื่อเชื่อมโยงข้อมูล (ฟังก์ชัน aggregate ใช้สำหรับการประมวลผลเพื่อรวบรวมข้อมูลให้เป็นกลุ่มเดียวกัน เช่น การหาผลรวม การหาค่าเฉลี่ย เป็นต้น)

Table 2 Displays examples of SQL Language commands and MongoDB Query Language commands designed for testing

| SQL Language command for MySQL | MongoDB Query Language command for MongoDB |
|--|--|
| SELECT * FROM tbarcode_location | db.tbarcode_location.find({}) |
| SELECT * FROM tbarcode WHERE amount_money > 2000 | db.tbarcode.find({ "amount_money": { \$gt: "2000" } }) |
| SELECT tbarcode.article_id, tbarcode.amount_money, tbarcode.article_name, tbarcode_location.move_date FROM tbarcode INNER JOIN tbarcode_location ON tbarcode.article_no = tbarcode_location.article_no | db.tbarcode.aggregate([{ \$lookup: { from: "tbarcode_location", localField: "article_no", foreignField: "articleno", as: "articlelocation" } }, { \$unwind: "\$articlelocation" }, { \$project: { "article_no": 1, "article_id": 1, "amount_money": 1, "article_name": 1, "move_date": "\$articlelocation.move_date" } }]) |

2.4 การพัฒนาเว็บแอปพลิเคชันเพื่อทดสอบประสิทธิภาพการสืบค้น

ผู้วิจัยได้ทำการพัฒนาเว็บแอปพลิเคชันจำลองเพื่อใช้ทดสอบประสิทธิภาพการสืบค้นข้อมูลครุภัณฑ์ จากฐานข้อมูล asset ของ MySQL และฐานข้อมูล asset_mongo ของ MongoDB โดยใช้ตารางและคอลเลคชันตามที่ได้กล่าวไว้ในตารางที่ 1 มีการสร้างคำสั่งและเพิ่มเงื่อนไขการสืบค้นในรูปแบบของ Drop-down list โดยแสดงเป็นตัวเลือกเพื่อช่วยในการค้นหาผ่านหน้าจอจัดรูปที่ 20 และรูปที่ 21

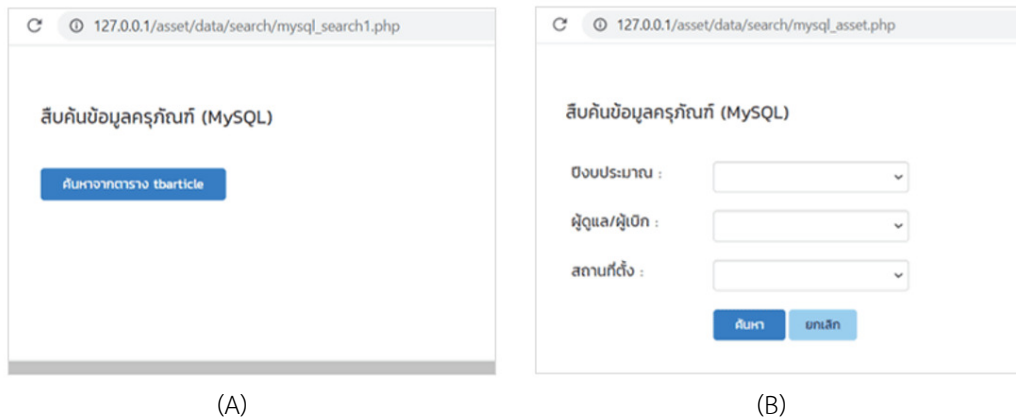


Figure 20 Web application for asset data retrieval from MySQL database
(A) Retrieval from 1 table (B) Retrieval from 4 tables with conditions

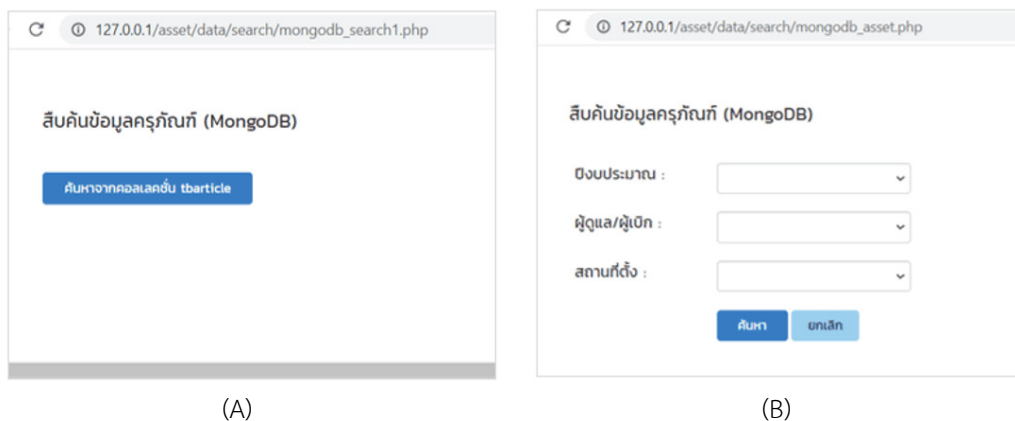


Figure 21 Web application for asset_mongo data retrieval from MongoDB database
(A) Retrieval from 1 collection (B) Retrieval from 4 collections with conditions

2.5 การทดสอบและการรวบรวมข้อมูลการสืบค้น

การทดสอบและการรวบรวมข้อมูลการสืบค้นจากฐานข้อมูลทั้งสองประเภทจะใช้วิธีการดังนี้

1) บันทึกเวลาในการคิวรีคำสั่งแต่ละคำสั่งที่ได้ออกแบบไว้เป็นจำนวน 10 ครั้งและหาค่าเฉลี่ย

2) บันทึกเวลาในการสืบค้นข้อมูลผ่านเว็บแอปพลิเคชันตามเงื่อนไขที่กำหนดไว้เป็นจำนวน 10 ครั้ง

และหาค่าเฉลี่ย

ผลการวิจัย

ผลที่ได้จากการทดสอบเปรียบเทียบประสิทธิภาพการสืบค้นข้อมูลระหว่างฐานข้อมูล MySQL และ MongoDB จากคำสั่งและจากเว็บแอปพลิเคชันเป็นดังนี้

Table 3 Displays a comparison of search speed using commands of MySQL and MongoDB databases (in seconds)

| Retrieve data from | Commands and conditions | MySQL(s) | MongoDB (s) |
|--|--|----------|-------------|
| 1 Table / 1 Collection | Search for all asset withdrawals | 0.06 | 0.00 |
| 1 Table / 1 Collection with 1 condition | Search for all assets with a price higher than 2000 Baht | 0.03 | 0.00 |
| Join 2 Tables / 2 Collections | Search for all asset withdrawals, displaying asset code, asset key, amount, asset description, and withdrawal date | 12.62 | 0.80 |
| Join 3 Tables / 3 Collections with 1 condition | Search for all assets located in the Grand Meeting Room (201), displaying asset code, asset key, asset description, and location | 0.91 | 19.60 |
| Join 3 Tables / 3 Collections with 2 conditions | Search for all stool chairs located in the ITSC Corner room, displaying asset code, asset key, asset description, and location | 0.11 | 92.85 |

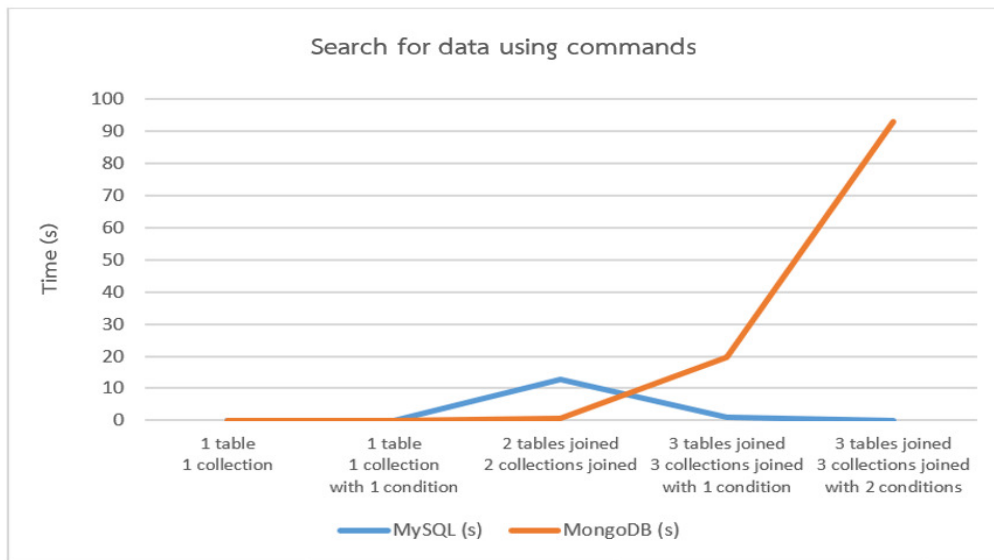


Figure 22 Line graph comparing search time using commands between MySQL and MongoDB databases

Table 4 Displays a comparison of search speed from web applications of MySQL and MongoDB databases (in seconds)

| Retrieve data from | Conditions | MySQL(s) | MongoDB (s) |
|---|--|----------|-------------|
| 1 Table / 1 Collection | Search for all asset data | 0.08 | 0.13 |
| | - Condition 1: Search for assets purchased in fiscal year 2560 | 0.66 | 29.78 |
| Join | - Condition 2: All assets managed by Mr. Wichai | 0.42 | 2.19 |
| | - Condition 3: Search for assets purchased in fiscal year 2556 and located in LAB A | 0.03 | 0.35 |
| 4 Tables / 4 Collections Selectable conditions** | - Condition 4: Search for assets purchased in fiscal year 2562 managed by Mr. Phanom | 1.16 | 12.30 |

** ในการทดสอบผู้วิจัยได้กำหนดเงื่อนไขโดยเลือกจาก Drop-down list ในเว็บแอปพลิเคชัน

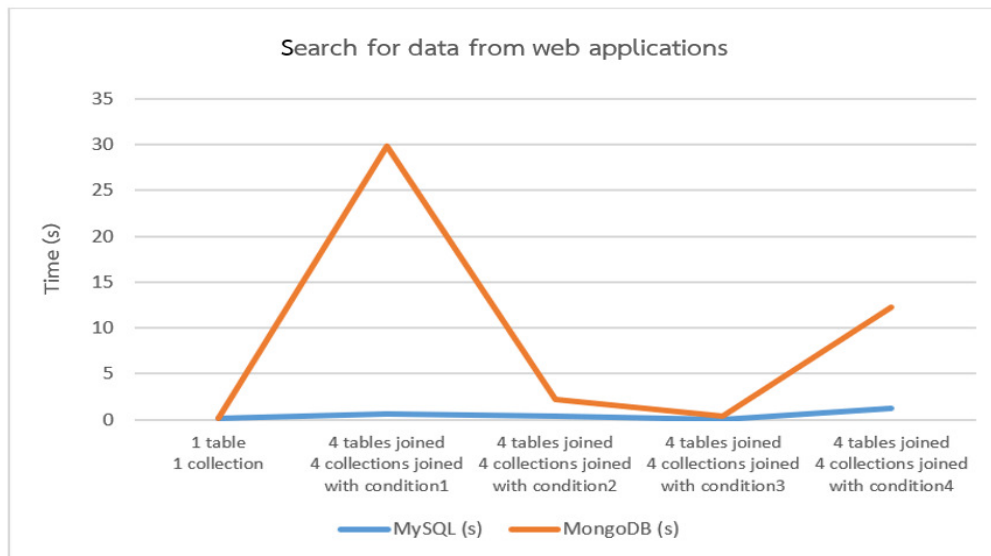


Figure 23 Line graph comparing search time from web applications between MySQL and MongoDB databases

เมื่อนำข้อมูลที่ได้จากตารางที่ 3 และตารางที่ 4 มาสร้างเป็นแผนภูมิกราฟเส้นดังรูปที่ 22 และรูปที่ 23 ตามลำดับ จะเห็นว่าประสิทธิภาพการสืบค้นข้อมูลครุภัณฑ์จากฐานข้อมูล MySQL ใช้เวลาในการค้นหาและแสดงผลได้รวดเร็วกว่าการค้นหาด้วยฐานข้อมูล MongoDB เกือบทุกกรณี

เนื่องจากข้อมูลครุภัณฑ์ที่นำมาใช้ในการทดสอบมีปริมาณข้อมูลในจำนวนที่ไม่มากนัก และเพื่อเป็นการพิสูจน์ว่าฐานข้อมูล NoSQL สามารถจัดการกับข้อมูลที่มีขนาดใหญ่ได้ดีและการเขียนหรือสืบค้นข้อมูลทำได้รวดเร็ว ผู้วิจัยจึงได้ทำการทดสอบเพิ่มเติมโดยได้ดาวน์โหลดตัวอย่างฐานข้อมูล (Sample Database) ซึ่งเป็นฐานข้อมูลที่มีขนาดค่อนข้างใหญ่สำหรับการทดสอบจากเว็บไซต์ของ datacharmer/test_db (https://github.com/datacharmer/test_db) และทำการแปลงข้อมูลจากฐานข้อมูล MySQL มาไว้ใน MongoDB ข้อมูลที่ทดสอบได้แก่ ข้อมูลพนักงาน และข้อมูลเงินเดือนพนักงาน ซึ่งในการทดสอบจะใช้ 2 ตาราง และ 2 คอลเลคชัน แสดงดังตารางที่ 5

Table 5 Displays additional tables and collections used for testing of MySQL and MongoDB databases

| MySQL Database (employees) | | | MongoDB Database (employees_mongo) | | |
|----------------------------|-------------|-----------|------------------------------------|-------------|-----------|
| Table | Data stored | Rows | Table | Data stored | Documents |
| employees | Employee | 300,024 | employees | Employee | 300,024 |
| salaries | Emp. salary | 2,844,047 | salaries | Emp. salary | 2,844,047 |

โดยการทดสอบจะเปรียบเทียบเฉพาะการสืบค้นข้อมูลจาก 1 ตาราง และ 1 คอลเลคชัน ทั้งแบบไม่มีเงื่อนไขและแบบมีเงื่อนไข ผลจากการทดสอบความเร็วในการสืบค้นข้อมูลจากตารางและจากคอลเลคชัน employees และ salaries ด้วยคำสั่งแบบไม่มีเงื่อนไขแบบมี 1 เงื่อนไข และแบบมี 2 เงื่อนไข ตามลำดับ โดยเก็บบันทึกเวลาในการสืบค้นของแต่ละคำสั่งเป็นจำนวน 10 ครั้ง และหาค่าเฉลี่ยของเวลาในการสืบค้น ผลลัพธ์แสดงดังตารางที่ 6

Table 6 Displays a comparison of search speed using commands from the employees database of MySQL and the employees_mongo database of MongoDB (in seconds)

| Retrieve data from | Commands and conditions | MySQL(s) | MongoDB (s) |
|--|---|----------|-------------|
| 1 Table / 1 Collection with no condition | Search for all employee data | 0.35 | 0.01 |
| 1 Table / 1 Collection with no condition | Search for all employee salaries | 2.19 | 0.02 |
| 1 Table / 1 Collection with 1 condition | Search for employee data with salaries greater than 60000 | 1.79 | 0.02 |
| 1 Table / 1 Collection with 2 conditions | Search for employee data with salaries greater than 50000 but less than 80000 | 1.92 | 0.00 |

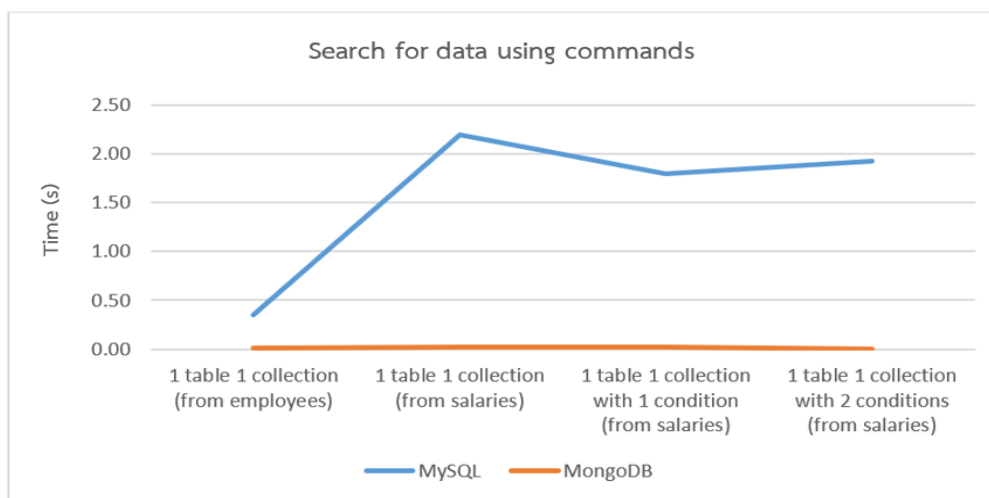


Figure 24 Line graph comparing search time using commands from the employees database of MySQL and the employees_mongo database of MongoDB

เมื่อนำข้อมูลที่ได้จากตารางที่ 6 มาสร้างเป็นแผนภูมิกราฟเส้นดังรูปที่ 24 จะเห็นว่า การสืบค้นข้อมูล โดยเฉพาะจาก 1 ตาราง และ 1 คอลเลคชัน ทั้งแบบไม่มีเงื่อนไขและแบบมีเงื่อนไข พบว่าการค้นหาข้อมูล จากฐานข้อมูล MongoDB ใช้เวลาค้นหาที่เร็วกว่าฐานข้อมูล MySQL ทุกกรณี โดยจะเห็นว่าเมื่อปริมาณ ข้อมูลที่ค้นหาเพิ่มขึ้นเวลาที่ใช้ในการค้นหาก็จะยิ่งน้อยลง

อภิปรายและสรุปผลการวิจัย

จากการวิเคราะห์ผลการเปรียบเทียบประสิทธิภาพการสืบค้นข้อมูลระหว่างฐานข้อมูลทั้งสองประเภท แสดงให้เห็นว่าถึงแม้ฐานข้อมูล NoSQL จะสามารถช่วยจัดการกับข้อมูลที่มีขนาดใหญ่ได้ดี การเขียนหรือการ สืบค้นข้อมูลทำได้รวดเร็ว แต่ไม่ได้หมายความว่าจะสามารถแปลงข้อมูลเชิงสัมพันธ์มาเป็นฐาน ข้อมูล NoSQL แล้วนำมาประยุกต์ใช้กับระบบสารสนเทศเพื่อให้ระบบสารสนเทศนั้นสามารถให้บริการอย่าง มีประสิทธิภาพได้ในทันที ในการศึกษาพบว่าการใช้คำสั่งเพื่อเข้าถึงข้อมูลของฐานข้อมูลเชิงสัมพันธ์ ด้วยวิธีการเชื่อมตารางจะส่งผลต่อประสิทธิภาพในการเข้าถึงข้อมูล ช่วยให้เข้าถึงข้อมูลได้รวดเร็วขึ้น สอดคล้องกับงานวิจัยของ ชยาพร แก่นสาร [1] และจะเห็นว่าสามารถนำฐานข้อมูล NoSQL มาใช้แทนฐานข้อมูล เชิงสัมพันธ์ และทำงานร่วมเว็บแอปพลิเคชันซึ่งเป็นระบบสารสนเทศที่ถูกจำลองขึ้นเพื่อใช้ในการทดสอบได้ แต่ ประสิทธิภาพ ในการสืบค้นข้อมูลของฐานข้อมูล NoSQL ในกรณีที่มีการเชื่อมข้อมูลระหว่างคอลเลคชันที่มีการ กำหนดเงื่อนไข การค้นหาจะทำได้ช้ากว่าฐานข้อมูลเชิงสัมพันธ์ สอดคล้องกับงานวิจัยของ อานันท์ เงินบำรุง [4] แต่แตกต่างจากงานวิจัยของ นรุทธิ์ บุตรพลอย และ พรหมเมศ วีระพันธ์ [5] ซึ่งผลการทดลองพบว่า ฐานข้อมูล NoSQL ใช้เวลาในการค้นหาข้อมูลที่เร็วกว่า เนื่องจากใช้ฐานข้อมูลที่มีโครงสร้างเป็นแบบ Embedded Document อย่างไรก็ตามการสืบค้นข้อมูลด้วยคำสั่งและสืบค้นข้อมูลจากเว็บแอปพลิเคชันจาก ฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูล NoSQL ผลลัพธ์ที่ได้จากการสืบค้นข้อมูลของสองเทคโนโลยีที่แตกต่าง กัน ต่างให้ผลลัพธ์ที่ถูกต้องแม่นยำและตรงกัน 100% สอดคล้องกับงานวิจัยของ รชต ทิมาสรวิชิก [3]

งานวิจัยนี้ได้ศึกษาแนวทางในการจัดเก็บข้อมูลของฐานข้อมูล NoSQL ซึ่งโครงสร้างการจัดเก็บข้อมูล จะมีลักษณะแตกต่างจากฐานข้อมูลเชิงสัมพันธ์ โดยการจัดเก็บข้อมูลของฐานข้อมูลเชิงสัมพันธ์จะใช้ MySQL ซึ่งมีรูปแบบในการจัดเก็บที่แน่นอน ข้อมูลจะถูกเก็บไว้ในตาราง มีการกำหนดชนิดของข้อมูลที่แน่นอน ข้อมูล แต่ละตารางประกอบด้วยแถวและคอลัมน์ การเชื่อมโยงข้อมูลระหว่างตารางจะใช้วิธีการเชื่อมตารางที่มีความ สัมพันธ์กันเข้าด้วยกัน ในขณะที่การจัดเก็บข้อมูลของฐานข้อมูล NoSQL จะใช้ MongoDB ซึ่งมีรูปแบบในการจัดเก็บ ที่ไม่แน่นอน การจัดเก็บข้อมูลไม่จำเป็นต้องกำหนดรูปแบบหรือประเภทของข้อมูล ข้อมูลจะถูกเก็บไว้ในคอลเลคชัน และในแต่ละคอลเลคชันจะประกอบด้วยดีออกคิวเมนต์ ซึ่งจะเก็บข้อมูลในรูปแบบของ Field-Value โดย สามารถเปรียบเทียบโครงสร้างการจัดเก็บข้อมูลของฐานข้อมูลทั้งสองประเภทดังตารางที่ 7

Table 7 Compares the structure of data storage between relational databases and NoSQL databases

| Relational Database (MySQL) | NoSQL Database (MongoDB) |
|-----------------------------|---|
| Database | Database |
| Table | Collection |
| Row/Record | Document/Object |
| Column | Field/Key |
| Table Join | Embedded Document, Referenced Document (with aggregate function) |

การเปรียบเทียบประสิทธิภาพความเร็วในการสืบค้นข้อมูลของงานวิจัยนี้สามารถสรุปได้ว่า การสืบค้นข้อมูลสำหรับระบบสารสนเทศกรณีศึกษาตัวอย่างชุดข้อมูลครุภัณฑ์ ด้วยฐานข้อมูลเชิงสัมพันธ์ มีประสิทธิภาพความเร็วในการสืบค้นที่เร็วกว่าการสืบค้นด้วยฐานข้อมูล NoSQL ซึ่งสาเหตุที่ทำให้ประสิทธิภาพการสืบค้นข้อมูลด้วยฐานข้อมูล NoSQL ค้นหาข้อมูลได้ช้ากว่า เนื่องมาจากโครงสร้างการจัดเก็บข้อมูลครุภัณฑ์ในฐานข้อมูล NoSQL ที่แปลงมาจากฐานข้อมูลเชิงสัมพันธ์มีโครงสร้างคือคิวเมนต์เป็นแบบ Referenced Document ซึ่งข้อมูลที่เกี่ยวข้องกันถูกแยกเก็บในคอลเลคชันที่เป็นอิสระจากกันและเชื่อมโยงกันด้วยคีย์อ้างอิง ดังนั้นในการค้นหาข้อมูลจึงต้องใช้ฟังก์ชันเพื่อเชื่อมโยงข้อมูลระหว่างคอลเลคชัน ทำให้ใช้เวลาในการประมวลผลมากขึ้น จึงค้นหาข้อมูลได้ช้ากว่า

อย่างไรก็ตามจากการศึกษาเพิ่มเติมโดยใช้ฐานข้อมูลตัวอย่างที่มีปริมาณข้อมูลมากกว่าข้อมูลครุภัณฑ์ถึง 100 เท่า พบว่าการสืบค้นข้อมูลโดยเฉพาะจาก 1 ตาราง และ 1 คอลเลคชัน ทั้งแบบไม่มีเงื่อนไขและแบบมีเงื่อนไข พบว่าฐานข้อมูล NoSQL ใช้เวลาในการค้นหาข้อมูลเร็วกว่าฐานข้อมูลเชิงสัมพันธ์

ข้อเสนอแนะสำหรับแนวทางในการแก้ปัญหาเพื่อเพิ่มประสิทธิภาพความเร็วในการสืบค้นข้อมูลของฐานข้อมูล NoSQL ที่จะนำมาใช้กับระบบบริหารจัดการครุภัณฑ์ในอนาคต คือ 1) ปรับแก้โครงสร้างการจัดเก็บข้อมูลจาก Referenced Document ไปเป็น Embedded Document เนื่องจาก Embedded Document จะเก็บข้อมูลที่เกี่ยวข้องทั้งหมดรวมไว้ในคิวเมนต์เดียวกันจึงค้นหาข้อมูลได้รวดเร็วกว่า อีกทั้งยังช่วยลดจำนวนคำสั่งที่ใช้ในการสืบค้นข้อมูล 2) ปรับแก้ไขโค้ดระบบบริหารจัดการครุภัณฑ์ที่พัฒนาด้วยภาษา PHP เพื่อให้สามารถติดต่อและจัดการฐานข้อมูล NoSQL ได้อย่างมีประสิทธิภาพ

ผลกระทบที่เกิดจากการปรับแก้โครงสร้างการจัดเก็บข้อมูล คือ ต้องใช้เวลาในการศึกษาและออกแบบคอลเลคชันหรือคิวเมนต์ เพื่อให้มั่นใจว่าการย้ายข้อมูลเป็นไปอย่างราบรื่นและมีประสิทธิภาพที่สุด รวมถึงต้องใช้เวลาในการเขียนคำสั่งเพื่อโอนย้ายข้อมูลจาก Referenced Document มาเป็น Embedded Document

ตัวอย่างเปรียบเทียบโครงสร้างการจัดเก็บข้อมูล และคำสั่งที่ใช้ในการสืบค้นข้อมูลของคิวเมนต์ทั้งสองประเภท แสดงดังรูปที่ 25 และรูปที่ 26

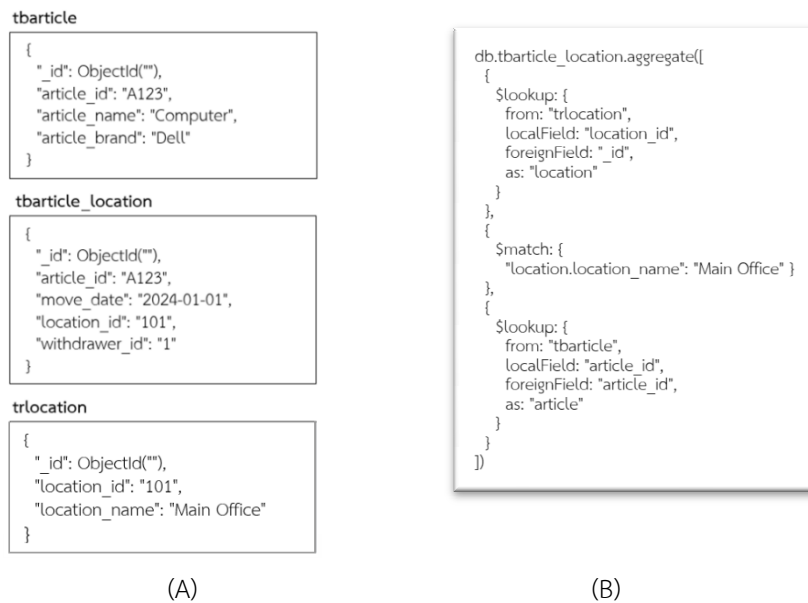


Figure 25 (A) Referenced Document (B) MongoDB query commands

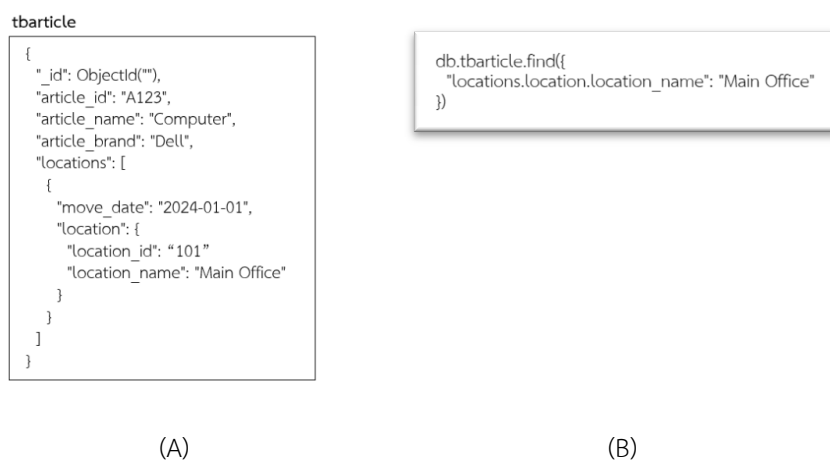


Figure 26 (A) Embedded Document (B) MongoDB query command

กิตติกรรมประกาศ

ขอขอบคุณมหาวิทยาลัยเชียงใหม่ในการสนับสนุนทุนวิจัยสถาบันเพื่อพัฒนามหาวิทยาลัยเชียงใหม่ ประจำปี พ.ศ. 2565 เพื่อใช้ในการทำวิจัยครั้งนี้

References

- [1] Kaensar, C. (2012). Analysis of the Query Plan for Assessing Optimizer Performance of SQL Select Statement. *The Journal of KMUTNB*, 22(3), 721-734. (in Thai)
- [2] MySQL. (2021). *MySQL 5.7 Reference Manual*. Available from <https://dev.mysql.com/doc/refman/5.7/en/language-structure.html>. Accessed date: 9 November 2021.
- [3] Timasornwichakit, R. (2017). *Comparative Study between Big Data Technology (Hadoop /MapReduce) and Relational Database Query System (MySQL) Case study: Healthcare Dataset*, (Master thesis, Dhurakij Pundit University). (in Thai)
- [4] Ngoenbumroong, A. (2020). *Performance Comparison Between Sql-Based & Document - Based Data Warehouse*, (Master thesis, Kasetsart University). (in Thai)
- [5] Butploy, N. & Verapan, P. (2024). Comparative Analysis of Database Management Approaches: Relational and NoSQL Databases - A Case Study of the Student Database at Kamphaeng Phet Rajabhat University. *Journal of Technology Management and Digital Innovation*, 1(1), 1-10. (in Thai)
- [6] Sumathi, S. & Esakkirajan, S. (2007). *Fundamentals of relational database management systems*. New York: Springer Berlin Heidelberg.
- [7] Jaratsantijit, Y. (2019). *Managing Data with MongoDB, a NoSQL Database*, Chiang Mai: Jarus Printing Co., Ltd. (in Thai)
- [8] Tiwari, S. (2011). *Professional NoSQL*. Indianapolis: John Wiley & Sons, Inc.
- [9] Tahaghoghi, S. & Williams, H. (2007). *Learning MySQL*. California: O'Reilly Media, Inc.
- [10] Navicat. (2021). *Manuals*. Available from <https://www.navicat.com/en/support/online-manual>. Accessed date: 22 December 2021.
- [11] MongoDB. (2022). *MongoDB Documentation*. Available from <https://www.mongodb.com/docs/manual/>. Accessed date: 15 January 2022.
- [12] Studio 3T. (2022). *Getting Started with Studio*. Available from <https://studio3t.com/knowledge-base/articles/getting-started/>. Accessed date: 26 January .2022
- [13] PHP. (2021). *PHP Manual*. Available from <https://www.php.net/manual/en/book.mysqli.php>. Accessed date: 25 November .2021
- [14] PHP. (2021). *Using the PHP Library for MongoDB (PHPLIB)*. Available from <https://www.php.net/manual/en/mongodb.tutorial.library.php>. Accessed date: 3 April .2022