

Received: 13 ก.พ. 2569

Revised: 20 พ.ค. 2569

Accepted: 16 ก.ค. 2569

การเปรียบเทียบกลยุทธ์การออกแบบโครงสร้างฐานข้อมูลเพื่อเพิ่มประสิทธิภาพการประมวลผลข้อมูล
การติดเชื้อขนาดใหญ่สำหรับเว็บแดชบอร์ด

A Comparative Analysis of Database Schema Design Strategies
for Optimizing Large-Scale Infection Data Processing in Web Dashboards

พิมลพรรณ ดวงดีบ, ทศพร ทานสุพรรณ, ก้องกาญจน์ ดุลยไชย,
อรรณวิทย์ ชังคมานนท์ และ พาสน์ ปราโมกษ์ชน*
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

Pimonpan Dountip, Thodsaporn Tansupan, Kongkarn Dullayachai,
Attawit Changkamanon and Part Pramokchon*

Computer Science Department, Faculty of Science, Maejo University

*Corresponding author: part@maejo.mju.ac.th

Abstract

This research aims to 1) investigate and compare the performance of seven database schema designs, and 2) evaluate the impact of data storage strategies on the performance of web dashboards for large-scale infectious disease data. The experiment utilized the PostgreSQL database management system, and the web application was developed using Next.js in conjunction with Prisma. The evaluation was conducted under identical conditions utilizing the same dataset, frontend codebase, and testing procedures across three main web pages with 30 repeated iterations per method. Performance was assessed using Google Lighthouse. Statistical analysis via One-way ANOVA revealed that Method F (Denormalized without Index and Partition) achieved the highest average Performance Score of 84.40 on the overview page, differing significantly from the other methods ($p < 0.001$). This was followed by Method G (Denormalized with Index without Partition) at 82.60; while Method A (Normalized with Index and Partition); and Method E (Materialized View) both scored 80.33. Meanwhile, the scores for Accessibility, Best

Practices, and Search Engine Optimization (SEO) remained similar across all methods. The results indicate that the database schema design significantly affects dashboard performance. These experimental findings, specific to the dataset context and environment studied, can serve as a preliminary guideline for designing large-scale infectious disease surveillance systems in the future.

Keywords *database schema design; web dashboard performance; denormalization; materialized view; Google Lighthouse*

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์ 1) เพื่อศึกษาและเปรียบเทียบประสิทธิภาพของโครงสร้างฐานข้อมูล 7 วิธี และ 2) เพื่อประเมินผลกระทบของแนวทางการจัดเก็บข้อมูลต่อประสิทธิภาพเว็บแดชบอร์ดเฝ้าระวังโรค โดยใช้ระบบจัดการฐานข้อมูล PostgreSQL และพัฒนาเว็บแอปพลิเคชันด้วย Next.js ร่วมกับ Prisma ภายใต้เงื่อนไขเดียวกันทั้งชุดข้อมูล โค้ดส่วนหน้า และขั้นตอนทดสอบบน 3 หน้าเว็บ พร้อมทดสอบซ้ำวิธีละ 30 รอบ และประเมินประสิทธิภาพด้วย Google Lighthouse ผลการวิเคราะห์ทางสถิติด้วย One-way ANOVA พบว่า วิธี F การทำ Denormalized โดยไม่ทำ Index และ Partition ให้คะแนนประสิทธิภาพ Performance Score สูงสุดเฉลี่ย 84.4 คะแนนในหน้าภาพรวม ซึ่งแตกต่างจากวิธีอื่นอย่างมีนัยสำคัญทางสถิติ ($p < 0.001$) รองลงมาคือ วิธี G การทำ Denormalized และ Index โดยไม่ทำ Partition ที่ 82.60 ส่วน วิธี A การทำ Normalized Index และ Partition และ วิธี E การทำ Materialized View ได้ 80.33 เท่ากัน ขณะที่ค่า Accessibility Best Practices และ SEO ใกล้เคียงกันทุกวิธี ผลการศึกษาแสดงให้เห็นว่าโครงสร้างฐานข้อมูลมีผลต่อประสิทธิภาพแดชบอร์ดอย่างมีนัยสำคัญ ทั้งนี้ผลการทดลองดังกล่าวอ้างอิงจากบริบทของชุดข้อมูลและสภาพแวดล้อมที่ใช้ศึกษา จึงสามารถนำผลลัพธ์ที่ได้ไปใช้เป็นแนวทางเบื้องต้นในการออกแบบระบบเฝ้าระวังโรคขนาดใหญ่ต่อไป

คำสำคัญ: *โครงสร้างฐานข้อมูล; ประสิทธิภาพระบบหน้าปัดแสดงผลข้อมูล; การลดรูปบรรทัดฐาน; Materialized view; Google Lighthouse*

1. บทนำ

ปัจจุบันการเฝ้าระวังโรคติดต่อต้องอาศัยระบบสารสนเทศที่แสดงผลได้รวดเร็ว ถูกต้อง และเข้าใจง่าย โดยเฉพาะโรคไข้หวัดใหญ่ซึ่งมีความสำคัญอย่างยิ่งต่องานเฝ้าระวังทางสาธารณสุข เนื่องจากมีสถานการณ์การระบาดที่เปลี่ยนแปลงอย่างรวดเร็วและมีปริมาณผู้ป่วยสะสมจำนวนมาก ดังเช่นระบบรายงานการระบาดที่มีการจัดเก็บข้อมูลผู้ป่วยไข้หวัดใหญ่สูงถึงประมาณ 693,184 ระเบียบ (กรมควบคุมโรค, 2567) ทั้งนี้ ปริมาณข้อมูลที่มหาศาลมักนำมาซึ่งปัญหาข้อขัดในการประมวลผล ซึ่งงานวิจัยหลายชิ้นชี้ให้เห็นว่าหากระบบขาดการออกแบบโครงสร้างฐานข้อมูลที่เหมาะสม จะส่งผลให้ระบบประมวลผลและตอบสนองล่าช้าอย่างมากเมื่อสืบค้นข้อมูลขนาดใหญ่ (Makris et al., 2021) อันจะส่งผลกระทบต่อตรงต่อการวิเคราะห์สถานการณ์และการตัดสินใจแก้ไขปัญหาทางสาธารณสุขได้อย่างไม่ทันทั่วทั้งที่ เว็บแดชบอร์ดจึงเป็นเครื่องมือสำคัญในการสื่อสารข้อมูลเชิงสรุปและสนับสนุนการตัดสินใจเว็บแดชบอร์ดจึงเป็นเครื่องมือสำคัญในการสื่อสารข้อมูลเชิงสรุปและสนับสนุนการตัดสินใจ (Ghazisaeidi et al., 2015; Ahamed, Khan, & Haque, 2021; Kukkadapu, 2025) อย่างไรก็ตามประสิทธิภาพของแดชบอร์ดไม่ได้ขึ้นกับส่วนติดต่อผู้ใช้เพียงอย่างเดียว แต่ขึ้นกับโครงสร้างฐานข้อมูลโดยตรง เนื่องจากคำสั่งสืบค้นมักมีการรวมผล การกรองหลายเงื่อนไข และการเชื่อมหลายตาราง ซึ่งอาจทำให้ระบบตอบสนองช้าหากออกแบบสคีมาไม่เหมาะสม

ในทางวิชาการ แม้ปัจจุบันจะมีแนวโน้มการเปลี่ยนผ่านไปสู่เทคโนโลยีฐานข้อมูลขั้นสูง เช่น Column-oriented Database หรือ In-memory Database เพื่อรองรับการวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data) แต่สำหรับหน่วยงานหรือองค์กรหลายแห่ง ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) แบบดั้งเดิมยังคงเป็นมาตรฐานที่คุ้นเคยและเข้าถึงได้ง่าย การประยุกต์ใช้เทคนิคการออกแบบโครงสร้าง เช่น การลดรูปบรรทัดฐาน (Denormalization) การสร้างดัชนี (Index) หรือการแบ่งพาร์ทิชัน (Partition) จึงยังคงเป็นแนวทางที่จำเป็นและเหมาะสมในการดัดแปลงภาพสูงสุดของระบบฐานข้อมูลเดิมที่มีอยู่ (Lakkimsetty, 2023; Mohsin, 2025) แนวคิดฐานข้อมูลเชิงสัมพันธ์เน้นความถูกต้อง งานวิจัยนี้จึงเปรียบเทียบโครงสร้างฐานข้อมูล 7 วิธี ภายใต้เงื่อนไขควบคุมเดียวกัน ได้แก่ ใช้ชุดข้อมูลเดียวกัน โค้ดส่วนแสดงผลเดียวกัน และกระบวนการทดสอบเดียวกัน โดยประเมินผลด้วย Google Lighthouse ทั้งคะแนนหมวดการประเมินและตัวชี้วัดเชิงเทคนิค เพื่อหาแนวทางออกแบบฐานข้อมูลที่เหมาะสมสำหรับเว็บแดชบอร์ดเฝ้าระวังโรคขนาดใหญ่ในสภาพแวดล้อมจริง

2. วัตถุประสงค์

ผู้วิจัยได้กำหนดวัตถุประสงค์ของการวิจัยออกเป็นข้อ ๆ เพื่อความสอดคล้องกับกระบวนการทดลองและการอภิปรายผล ดังนี้

1. เพื่อศึกษาและเปรียบเทียบประสิทธิภาพของการออกแบบสคีมาฐานข้อมูลทั้ง 7 วิธี (วิธี A-G) สำหรับเว็บแดชบอร์ดแสดงผลข้อมูลการระบาดโรคไข้หวัดใหญ่ ภายใต้การควบคุมชุดข้อมูลและส่วนติดต่อผู้ใช้ (User Interface) ให้เป็นไปในทิศทางเดียวกัน
2. เพื่อประเมินความแตกต่างด้านประสิทธิภาพการค้นคืนและการดึงข้อมูลของแต่ละแนวทางการจัดเก็บข้อมูล บนหน้าเว็บหลักจำนวน 3 หน้า โดยจำแนกการวัดผลผ่านเครื่องมือ Google Lighthouse ออกเป็น 2 ส่วนหลัก ได้แก่
 - คะแนนหมวดหมู่หลัก Performance Accessibility Best Practices และ SEO
 - ตัวชี้วัดเชิงเทคนิค (Technical Metrics): First Contentful Paint (FCP) Largest Contentful Paint (LCP) Speed Index (SI) Total Blocking Time (TBT) และ Cumulative Layout Shift (CLS)

3. งานวิจัยที่เกี่ยวข้อง

ในการพัฒนาระบบแดชบอร์ดเฝ้าระวังโรคที่รองรับข้อมูลขนาดใหญ่ โครงสร้างฐานข้อมูลมีผลโดยตรงต่อทั้งความถูกต้องของข้อมูลและความเร็วการตอบสนองของหน้าเว็บ งานที่เกี่ยวข้องยืนยันบทบาทของข้อบังคับฐานข้อมูลเชิงสัมพันธ์ต่อคุณภาพข้อมูล (ประภาส ทองรัก, 2552) ความเหมาะสมของ star schema ต่อการวิเคราะห์หลายมิติและความสำคัญของการเตรียมข้อมูลให้เหมาะกับงานประมวลผล (พัชรียา ทองพูล และคณะ, 2562) ขณะที่แนวคิด dimensional modeling สนับสนุนการออกแบบเชิงมิติเพื่อการสรุปผลที่มีประสิทธิภาพ (Kimball & Ross, 2013) ซึ่งสอดคล้องกับ วิธี D การทำ Denormalized Index และ Partition วิธี F การทำ Denormalized โดยไม่ทำ Index และ Partition และ วิธี G การทำ Denormalized และ Index โดยไม่ทำ Partition

ในเชิงระบบจัดการฐานข้อมูล กลยุทธ์การปรับแต่งและเพิ่มประสิทธิภาพ (Database Optimization Strategies, 2023) ระบุว่าเทคนิคอย่าง partition pruning ช่วยลดช่วงข้อมูลที่ต้องสแกนเมื่อเงื่อนไขค้นหาตรงกับคีย์พาร์ทิชัน จึงรองรับแนวทางของ วิธี A การทำ Normalized Index และ Partition และ วิธี D การทำ Denormalized Index และ Partition อีกทั้งการใช้ materialized view ช่วยเก็บผลลัพธ์ควิรีเพื่อเรียกใช้ซ้ำและ refresh ได้เมื่อข้อมูลเปลี่ยน ซึ่งสอดคล้องกับ วิธี E การทำ Materialized View ส่วนงานด้านการสร้างระบบประมวลผลข้อมูลประสิทธิภาพสูงชี้ว่า workload เชิง

อ่านจะได้ประโยชน์อย่างมากจากการออกแบบที่ช่วยลดการอ่านข้อมูลที่ไม่จำเป็นลงได้ (Building High-Performance OLTP Systems, 2025)

นอกจากนี้ ในมิติของการนำข้อมูลมาแสดงผลผ่านเว็บแดชบอร์ด งานวิจัยยืนยันว่าประเด็นสำคัญในทางปฏิบัติของการพัฒนาแดชบอร์ดในภาคส่วนสาธารณสุขคือการจัดการความรวดเร็วในการประมวลผลตัวชี้วัดเพื่อการตัดสินใจ (Ghazisaeidi et al., 2015) ซึ่งสอดคล้องกับการออกแบบท่อส่งข้อมูลแบบปรับปรุงให้กระชับ (Streamlined Data Pipeline) เพื่อส่งมอบข้อมูลปริมาณมหาศาลขึ้นสู่หน้าจอแบบเวลาจริงได้อย่างมีประสิทธิภาพ (Kukkadapu, 2025)

ดังนั้น งานวิจัยนี้จึงเปรียบเทียบอย่างเป็นระบบครบทั้ง วิธี A การทำ Normalized Index และ Partition วิธี B การทำ Normalized และ Index โดยไม่ทำ Partition วิธี C การทำ Normalized โดยไม่ทำ Index และ Partition วิธี D การทำ Denormalized Index และ Partition วิธี E การทำ Materialized View วิธี F การทำ Denormalized โดยไม่ทำ Index และ Partition และ วิธี G การทำ Denormalized และ Index โดยไม่ทำ Partition ภายใต้เงื่อนไขควบคุมเดียวกัน เพื่อหาสคีมาที่ให้สมดุลเหมาะสมที่สุดต่อเว็บแดชบอร์ดข้อมูลผู้ป่วยขนาดใหญ่

4. วิธีการดำเนินการ

การวิจัยนี้เป็นการวิจัยเชิงทดลองเพื่อเปรียบเทียบประสิทธิภาพของโครงสร้างฐานข้อมูลสำหรับระบบเว็บแดชบอร์ดเฝ้าระวังโรค โดยมุ่งประเมินผลกระทบของรูปแบบฐานข้อมูลต่อประสิทธิภาพการแสดงผลหน้าเว็บในมุมมองผู้ใช้งาน ภายใต้เงื่อนไขการทดลองที่ควบคุมให้เหมือนกันทุกวิธี รายละเอียดวิธีดำเนินการวิจัยมีดังต่อไปนี้

4.1 กรอบแนวคิดการวิจัยเชิงทดลองเพื่อเพิ่มประสิทธิภาพฐานข้อมูล

งานวิจัยนี้ใช้กรอบการวิจัยเชิงทดลองร่วมกับแนวคิดวิศวกรรมประสิทธิภาพฐานข้อมูล เพื่อศึกษาว่าการออกแบบโครงสร้างฐานข้อมูลที่แตกต่างกันส่งผลต่อประสิทธิภาพหน้าเว็บแดชบอร์ดอย่างไร มีลักษณะการทำงานแบบโต้ตอบ (interactive dashboard) และครอบคลุมหน้าใช้งานหลัก 3 ส่วน ได้แก่

1. หน้าข้อมูลภาพรวม สำหรับแสดงภาพรวมสถานการณ์ผู้ป่วยระดับรวม
2. หน้าข้อมูลรายจังหวัด สำหรับวิเคราะห์สถานการณ์เชิงพื้นที่ระดับจังหวัด
3. หน้าเปรียบเทียบข้อมูลระหว่างจังหวัด สำหรับเปรียบเทียบแนวโน้มและความแตกต่างของตัวชี้วัดระหว่างพื้นที่

โดยในการทดลองเชิงเปรียบเทียบครั้งนี้ ผู้วิจัยได้กำหนดสมมติฐานการวิจัยทางสถิติไว้ดังนี้ โดยตั้งสมมติฐานหลัก (H_0): วิธีการจัดโครงสร้าง DB ทั้ง 7 แบบ ไม่ส่งผลต่อความเร็วในการโหลด ($\mu_A = \mu_B = \dots = \mu_G$) ที่ระดับนัยสำคัญ (α): 0.05 ซึ่งหากค่า p-value < 0.05 แสดงว่าปฏิเสธสมมติฐานหลัก ซึ่งหมายถึงมีอย่างน้อย 1 วิธีที่มีประสิทธิภาพต่างจากวิธีการอื่นอย่างชัดเจน

4.2 ข้อมูลและการเตรียมข้อมูล

ข้อมูลที่ใช้ในการทดลองเป็นข้อมูลผู้ป่วยโรคไข้หวัดใหญ่จากระบบรายงานการระบาด ประกอบด้วยตัวแปรสำคัญ 13 ตัวแปร เช่น วันที่เริ่มป่วย จังหวัด อายุ เพศ อาชีพ และรหัสโรค ข้อมูลจำนวน 693,184 ระเบียบ ในเชิงนิยามของ Big Data ยุคใหม่ ข้อมูลระดับแสนระเบียบอาจยังไม่ถือเป็นข้อมูลขนาดใหญ่มาก (Volume) แต่ในบริบทของระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ข้อมูลจำนวน 693,184 ระเบียบ ถือว่ามีปริมาณเพียงพอที่จะทำให้เกิดปัญหาคอขวด (Bottleneck) เมื่อต้องใช้คำสั่งสืบค้นที่มีความซับซ้อน (Complex Queries) เช่น การทำ Aggregation หลายมิติ หรือการ Join ข้ามตาราง ซึ่งเป็นสถานการณ์จริงที่พบในระบบเฝ้าระวังโรค จึงมีความเหมาะสมที่จะใช้เป็นชุดข้อมูลตัวแทนสำหรับการศึกษาประสิทธิภาพ

ก่อนการทดลอง ผู้วิจัยดำเนินการทำความสะอาดข้อมูล (Data Cleaning) และคัดกรองระเบียบที่ขาดข้อมูลสำคัญต่อการวิเคราะห์ (เช่น ไม่มีจังหวัดหรือวันที่เริ่มป่วย) ออกจากชุดทดลอง โดยมีจำนวนข้อมูลที่ถูกต้องออกทั้งสิ้น 26 ระเบียบ คิดเป็นสัดส่วนร้อยละ 0.04 ของข้อมูลทั้งหมด ซึ่งในทางสถิติถือเป็นสัดส่วนที่น้อยมากจนไม่มีนัยสำคัญ จึงไม่ส่งผลกระทบต่อความสมบูรณ์ของข้อมูลและไม่ก่อให้เกิดความลำเอียงของชุดข้อมูล (Dataset Bias)

นอกจากนี้ เพื่อให้เห็นภาพการประเมินประสิทธิภาพฐานข้อมูล (Database Performance) ที่ชัดเจน การทดสอบบน 3 หน้าเว็บหลักได้จำลองภาระงาน (Workload) ผ่านการกำหนดเงื่อนไขการค้นหาลงได้แก่ การกรองโรค (ไข้หวัดใหญ่) และช่วงเวลา 6 เดือน (01/01/2567 ถึง 30/06/2567) โดยแต่ละหน้ามีจำนวนคำสั่งและระดับความซับซ้อน (Complexity Level) แตกต่างกัน ดังนี้

1. หน้าข้อมูลภาพรวม (Dashboard) มีระดับความซับซ้อนปานกลาง (Medium Complexity) ใช้คำสั่งสืบค้นจำนวน 10 คำต่อหน้า โดยใช้คำสั่งประเภท Aggregation ร่วมกับ Filter เพื่อรวมผลยอดผู้ป่วยตามช่วงเวลาและโรคที่กำหนด

2. หน้าข้อมูลรายจังหวัด (ProvincialInfo): มีระดับความซับซ้อนปานกลาง (Medium Complexity) ใช้คำสั่งสืบค้นจำนวน 13 คำต่อหน้า โดยใช้รูปแบบการดึงข้อมูลและแสดงกราฟคล้าย

หน้า Dashboard แต่มีการเพิ่มเงื่อนไขคำสั่ง Filter แบบเจาะจงพื้นที่ เช่น ค้นหาเฉพาะจังหวัด กรุงเทพมหานคร ร่วมกับการทำ Aggregation

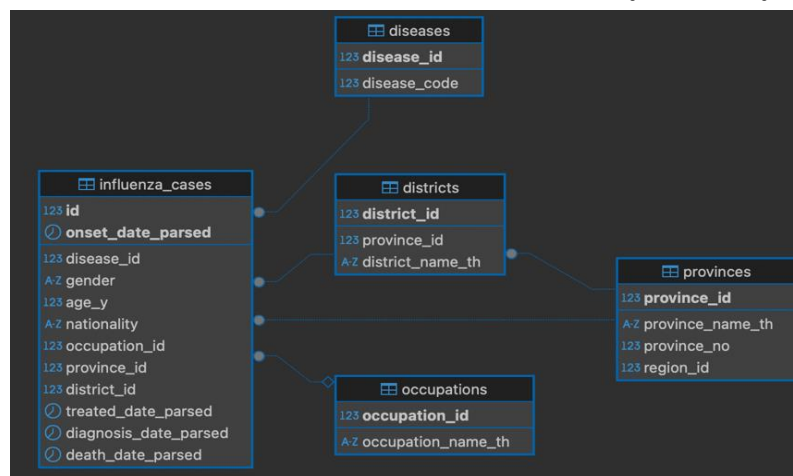
3. หน้าเปรียบเทียบข้อมูล (CompareInfo): มีระดับความซับซ้อนสูงสุด (High Complexity) ใช้คำสั่งสืบค้นจำนวน 13 คิวรีต่อหน้า เป็นการประมวลผลเพื่อเปรียบเทียบตัวชี้วัดระหว่าง 2 จังหวัด เช่น กรุงเทพมหานคร เทียบกับ เชียงใหม่ ภายใต้ช่วงเวลาเดียวกัน จึงต้องใช้คำสั่งประเภท Join หลายตารางย่อย ผสมกับการใช้ Filter ที่ซับซ้อนขึ้น และการทำ Aggregation ข้อมูลสองชุดพร้อมกัน

4.3 รูปแบบโครงสร้างฐานข้อมูลที่เปรียบเทียบ

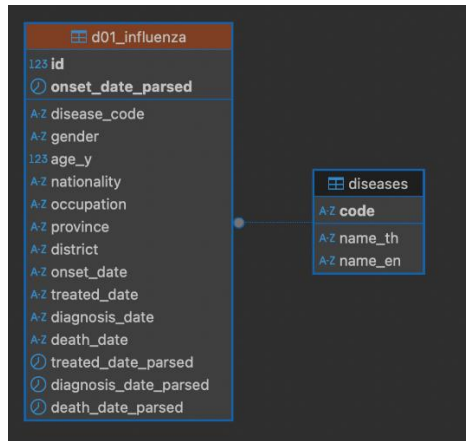
การวิจัยนี้เปรียบเทียบโครงสร้างฐานข้อมูล 7 วิธี เพื่อครอบคลุมแนวทางการออกแบบที่ต่างกัันทั้งเชิงตรรกะ (Logical Design) และเชิงกายภาพ (Physical Design) ดังนี้

1. วิธี A การทำ Normalized Index และ Partition
2. วิธี B การทำ Normalized และ Index โดยไม่ทำ Partition
3. วิธี C การทำ Normalized โดยไม่ทำ Index และ Partition
4. วิธี D การทำ Denormalized Index และ Partition
5. วิธี E การทำ Materialized View
6. วิธี F การทำ Denormalized โดยไม่ทำ Index และ Partition
7. วิธี G การทำ Denormalized และ Index โดยไม่ทำ Partition

เพื่อแสดงให้เห็นถึงความแตกต่างของการออกแบบเชิงตรรกะอย่างเป็นรูปธรรม ผู้วิจัยได้จัดกลุ่มโครงสร้างตารางพื้นฐานออกเป็น 2 รูปแบบหลัก ได้แก่ โครงสร้างแบบ Normalized และโครงสร้างแบบ Denormalized ดังแผนภาพความสัมพันธ์ของเอนทิตี (ER Diagram) ในรูปที่ 1 และรูปที่ 2 ตามลำดับ



รูปที่ 1 แผนภาพ ER Diagram แสดงโครงสร้างฐานข้อมูลแบบ Normalized



รูปที่ 2 แผนภาพ ER Diagram แสดงโครงสร้างฐานข้อมูลแบบ Denormalized

จากรูปที่ 1 โครงสร้างแบบ Normalized จะยึดหลักการลดความซ้ำซ้อนของข้อมูล (Data Redundancy) โดยแยกเก็บข้อมูลหลักของผู้ป่วย (ตาราง influenza_cases) ออกจากข้อมูลอ้างอิงเชิงมิติ (Dimension Tables) เช่น ตารางจังหวัด (provinces) อำเภอ (districts) และอาชีพ (occupations) ซึ่งแม้จะดีต่อการประหยัดพื้นที่จัดเก็บ แต่ในการเรียกดูข้อมูลบนแดชบอร์ด ระบบจำเป็นต้องใช้คำสั่ง JOIN หลายตารางเพื่อดึงข้อมูลมาแสดงผล

ในทางตรงกันข้าม รูปที่ 2 แสดงโครงสร้างแบบ Denormalized ซึ่งยุบรวมฟิลด์ข้อมูลอ้างอิงที่จำเป็นต้องใช้บ่อย (เช่น ชื่อจังหวัด รหัสโรค) มารวมไว้ในตารางแฟกต์หลักเพียงตารางเดียว (เช่น ตาราง d01_influenza) การออกแบบเชิงตรรกะลักษณะนี้ ช่วยขจัดภาระการประมวลผล (Overhead) ที่เกิดจากการ JOIN ตาราง ทำให้การสืบค้นและรวมผล (Aggregation) ของวิธีในกลุ่มนี้ (โดยเฉพาะวิธี F ที่ไม่ถูกลดทอนความเร็วจาก Overhead ของการทำ Index และ Partition) สามารถสแกนข้อมูลจากตารางเดียวและคืนผลลัพธ์ได้อย่างรวดเร็วที่สุด

สำหรับการออกแบบเชิงกายภาพ ในวิธีที่มีการทำดัชนี (Index) ผู้วิจัยเลือกใช้ดัชนีประเภท B-Tree ซึ่งมีประสิทธิภาพสูงในการค้นหาข้อมูลแบบช่วงเวลา (Range Query) และในวิธีที่มีการแบ่งพาร์ทิชัน (Partition) ผู้วิจัยใช้กลยุทธ์แบบ Range Partitioning โดยแบ่งตามช่วงเดือน/ปี เพื่อลดปริมาณข้อมูลที่ต้องสแกน กรอบการเปรียบเทียบทั้งหมดนี้จะช่วยให้สามารถวิเคราะห์ผลของการทำ Normalization และ Denormalization รวมถึงผลของการสรุปข้อมูลล่วงหน้าด้วย Materialized View ต่อประสิทธิภาพหน้าเว็บแดชบอร์ดได้อย่างเป็นระบบ

4.4 สภาพแวดล้อมและเครื่องมือทดสอบ

เพื่อให้ผลการทดลองมีความเที่ยงตรง ผู้วิจัยกำหนดให้ใช้สภาพแวดล้อมการทดสอบเดียวกันในทุกวิธีทดลอง และใช้เครื่องมือชุดเดียวกันตลอดการเก็บผล โดยมีองค์ประกอบดังนี้

- หน่วยประมวลผลกลาง (CPU): Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz
- หน่วยความจำ (RAM): 32 GB
- หน่วยความจำสำรอง (Storage): SSD รุ่น MSI M450 ความจุ 500 GB
- ระบบฐานข้อมูล: PostgreSQL 15.13

ทั้งนี้การศึกษานี้มุ่งเน้นการวัดผลกระทบในมุมมองที่ส่งผลต่อประสบการณ์ผู้ใช้ปลายทาง (User-centric metrics) ผ่านเครื่องมือ Google Lighthouse เป็นหลัก โดยครอบคลุมตั้งแต่กระบวนการดึงข้อมูลไปจนถึงการแสดงผล (End-to-End Performance) ซึ่งสะท้อนความเร็วที่ผู้ใช้งานสัมผัสได้จริงบนเว็บเบราว์เซอร์

4.5 ตัวแปรวิจัยและการควบคุมการทดลอง

การวิจัยนี้กำหนดให้ รูปแบบโครงสร้างฐานข้อมูล 7 วิธี เป็นตัวแปรต้น และกำหนดให้ ค่าตัวชี้วัดประสิทธิภาพหน้าเว็บจาก Google Lighthouse เป็นตัวแปรตาม เพื่อให้ผลการเปรียบเทียบมีความเที่ยงตรง ผู้วิจัยได้ควบคุมปัจจัยสำคัญให้คงที่ในทุกวิธีทดลอง ได้แก่ การใช้ชุดข้อมูลเดียวกัน โค้ดโปรแกรมประยุกต์ส่วนหน้าเดียวกัน ขั้นตอนและลำดับการทดสอบเดียวกัน และสภาพแวดล้อมการทดสอบเดียวกัน

ผู้วิจัยทำการทดสอบซ้ำจำนวน 30 ครั้ง (n=30) ในแต่ละรูปแบบจำลอง ซึ่งเป็นไปตามมาตรฐานการทดสอบประสิทธิภาพ จำนวนรอบการทดลองที่เพียงพอนี้ช่วยลดผลกระทบจากค่าความผิดปกติ และสร้างความน่าเชื่อถือทางสถิติตามทฤษฎีขีดจำกัดส่วนกลาง

4.6 ตัวชี้วัด Lighthouse เกณฑ์การแปลผล และสมการคำนวณ

ผู้วิจัยได้เลือกใช้ตัวชี้วัด Performance Score Largest Contentful Paint (LCP) Total Blocking Time (TBT) และ Speed Index (SI) สำหรับการศึกษาครั้งนี้ เพื่อให้สามารถประเมินประสิทธิภาพของระบบได้อย่างครอบคลุมในมิติที่ส่งผลกระทบต่อประสบการณ์ผู้ใช้งานจริง ทั้งนี้ผู้วิจัยได้พิจารณาตัดตัวชี้วัด Accessibility Best Practices SEO FCP และ CLS ออกจากการวิเคราะห์ เนื่องจากตัวชี้วัดกลุ่มนี้เป็นผลลัพธ์ที่ขึ้นอยู่กับการออกแบบส่วนแสดงผลและการตั้งค่าเซิร์ฟเวอร์เป็นหลัก ซึ่งไม่มีความสัมพันธ์ทางตรงกับความเร็วของโครงสร้างฐานข้อมูล ทำให้คะแนนของตัวชี้วัดกลุ่มนี้มีความคงที่และไม่สะท้อนผลจากการเปลี่ยนแปลงตัวแปรต้นของการวิจัย

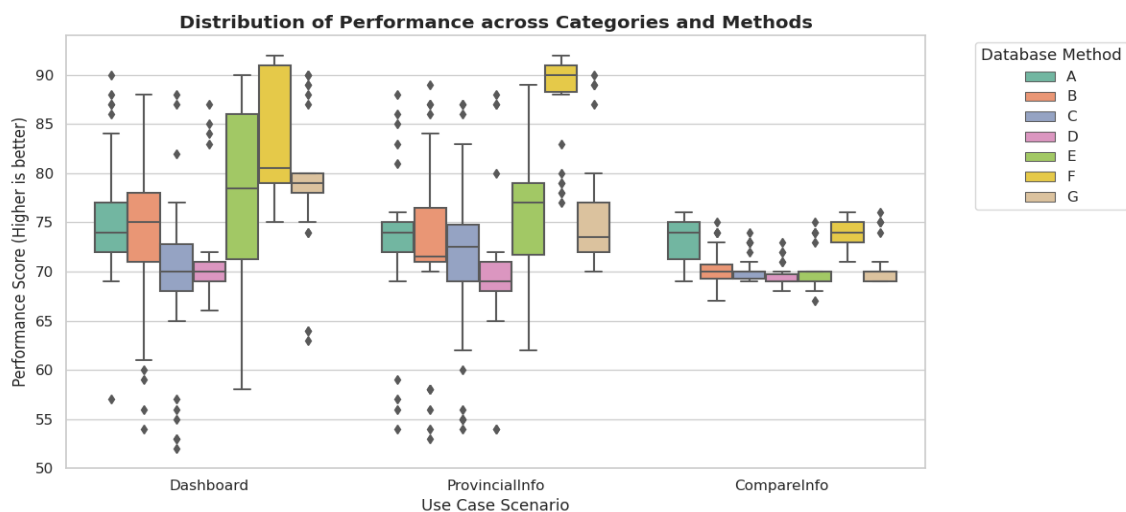
5. ผลการศึกษา

การศึกษานี้เป็นการประเมินประสิทธิภาพของเว็บเบราว์เซอร์ไฟร์ฟอกซ์ ภายใต้โครงสร้างฐานข้อมูลที่แตกต่างกันจำนวน 7 วิธี โดยพิจารณาหน้าเว็บตามลักษณะการใช้งานที่มีระดับความซับซ้อน

แตกต่างกันจำนวน 3 หน้า ได้แก่ หน้าข้อมูลภาพรวม (Dashboard) หน้าข้อมูลรายจังหวัด (ProvincialInfo) และหน้าเปรียบเทียบข้อมูลระหว่างจังหวัด (CompareInfo) ด้วยเครื่องมือ Google Lighthouse ผลการทดลองรายงานในรูปแบบค่าเฉลี่ยร่วมกับส่วนเบี่ยงเบนมาตรฐานจากการทดสอบซ้ำ 30 รอบ พร้อมทั้งวิเคราะห์ความแตกต่างเชิงสถิติด้วย One-way ANOVA และวิเคราะห์รายคู่ด้วย Tukey's HSD test ที่ระดับนัยสำคัญ 0.05 ดังนี้

ตารางที่ 1 เปรียบเทียบคะแนน Performance Score ใน 3 หน้าเว็บ (หน่วย: คะแนน 0-100, ค่าสูงแสดงถึงประสิทธิภาพที่ดีกว่า)

โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
A (Norm+Idx+Part)	75.6 ± 7.0	73.0 ± 8.0	73.4 ± 2.3
B (Norm+Idx)	74.0 ± 8.7	72.0 ± 9.9	70.4 ± 2.0
C (Norm, Baseline)	68.9 ± 9.0	70.9 ± 9.4	70.4 ± 1.4
D (Denorm+Idx+Part)	71.7 ± 5.5	70.4 ± 7.4	69.4 ± 1.3
โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
E (Materialized View)	78.0 ± 8.7	76.0 ± 7.1	70.2 ± 1.9
F (Denorm)	84.4 ± 5.9	87.7 ± 5.0	73.8 ± 1.7
G (Denorm+Idx)	79.3 ± 7.2	75.7 ± 5.9	70.5 ± 2.1
p-value (ANOVA)	< 0.001	< 0.001	< 0.001

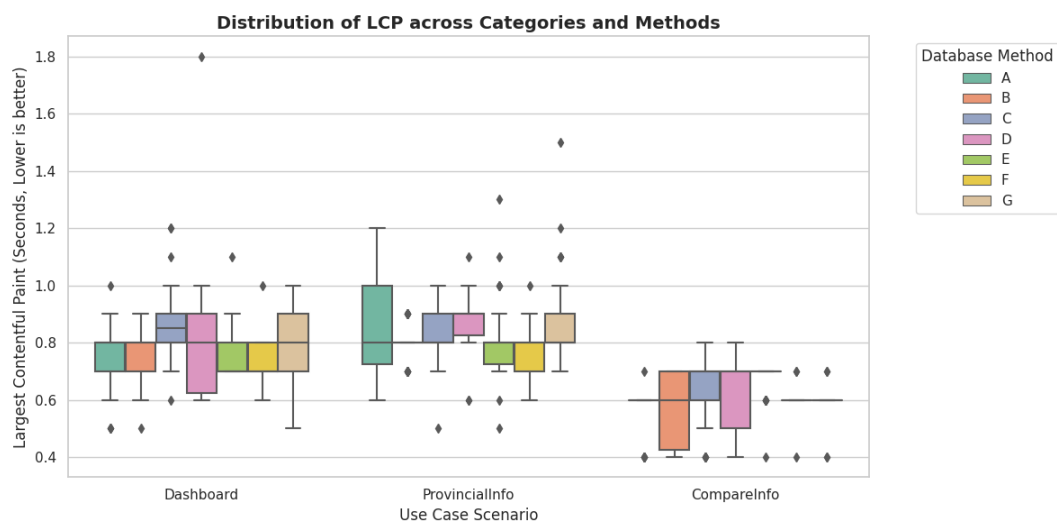


รูปที่ 3 แผนภาพ Boxplot แสดงการกระจายตัวของคะแนน Performance Score
แยกตามกรณีการใช้งาน

จากการทดสอบด้วย ANOVA พบว่าโครงสร้างฐานข้อมูลส่งผลต่อคะแนนประสิทธิภาพอย่างมีนัยสำคัญ ($p < 0.001$) ในทุกสถานการณ์ เมื่อทดสอบความแตกต่างรายคู่พบว่า การทำ Denormalization วิธี F ให้คะแนนสูงกว่าแบบ Normalization วิธี C อย่างมีนัยสำคัญทางสถิติ ($p < 0.001$) อย่างไรก็ตาม การเพิ่ม Partition ลงในโครงสร้างแบบ Denormalized วิธี D กลับทำให้คะแนนลดลงกว่าการไม่ใส่ วิธี G อย่างมีนัยสำคัญ ($p = 0.002$)

ตารางที่ 2 เปรียบเทียบระยะเวลา Largest Contentful Paint (LCP)(หน่วย: วินาที, ค่ายิ่งต่ำแสดงถึงประสิทธิภาพที่ดีกว่า)

โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
A (Norm+Idx+Part)	0.76 ± 0.13	0.84 ± 0.15	0.56 ± 0.09
B (Norm+Idx)	0.75 ± 0.09	0.80 ± 0.07	0.57 ± 0.12
C (Norm, Baseline)	0.86 ± 0.14	0.82 ± 0.10	0.62 ± 0.12
D (Denorm+Idx+Part)	0.81 ± 0.23	0.89 ± 0.11	0.61 ± 0.13
โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
E (Materialized View)	0.75 ± 0.09	0.82 ± 0.15	0.67 ± 0.07
F (Denorm)	0.73 ± 0.08	0.78 ± 0.08	0.60 ± 0.05
G (Denorm+Idx)	0.80 ± 0.11	0.86 ± 0.18	0.59 ± 0.06
p-value (ANOVA)	0.0049	0.0130	0.0004

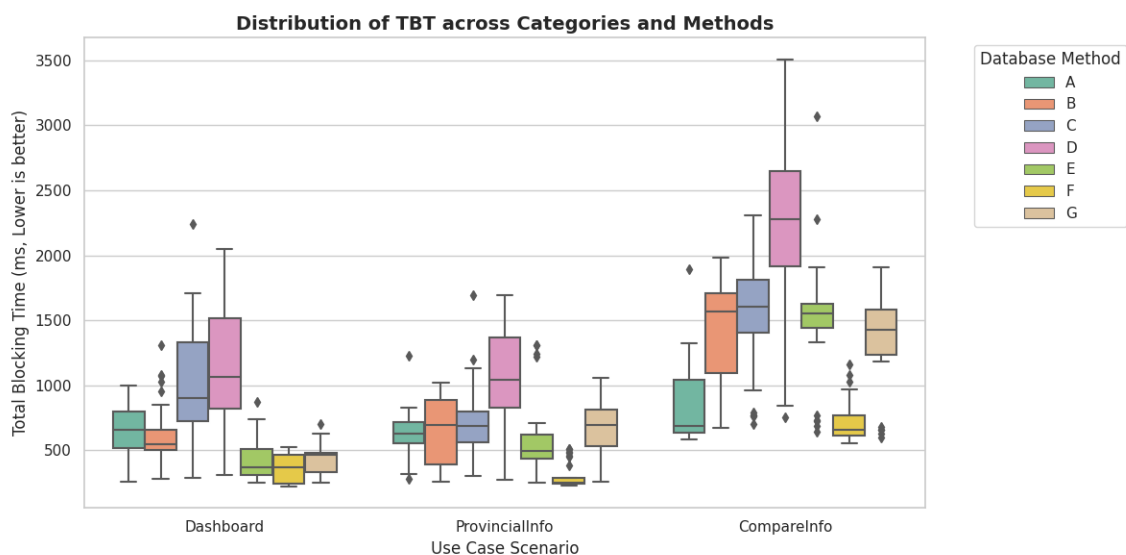


รูปที่ 4 Boxplot แสดงการกระจายของ LCP ในหน้า Dashboard แต่ละกรณี (Use Case Scenario)

ผลจาก ANOVA ชี้ว่าวิธีต่างๆ ให้ความเร็ว LCP แตกต่างกัน ($p < 0.05$) โดยวิธี F โหลดภาพหลักได้เร็วกว่าวิธี C ในหน้า Dashboard อย่างมีนัยสำคัญ ($p = 0.008$) แต่ในกรณีที่ Query มีความซับซ้อนสูง วิธี A สามารถจัดการดึง LCP ได้เร็วกว่าวิธี E การทำ Materialized View อย่างมีนัยสำคัญ ($p = 0.0003$) แสดงให้เห็นว่า Materialized View ไม่เหมาะกับคำสั่งคิวรีที่เปลี่ยนเงื่อนไขบ่อย

ตารางที่ 3 เปรียบเทียบระยะเวลา Total Blocking Time (TBT) (หน่วย: มิลลิวินาที, ค่าต่ำแสดงถึงประสิทธิภาพที่ดีกว่า)

โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
A (Norm+Idx+Part)	634 ± 218	614 ± 181	833 ± 315
B (Norm+Idx)	616 ± 254	674 ± 265	1425 ± 419
C (Norm, Baseline)	982 ± 473	716 ± 294	1524 ± 450
D (Denorm+Idx+Part)	1120 ± 502	1054 ± 412	2164 ± 718
โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
E (Materialized View)	436 ± 169	577 ± 302	1508 ± 482
F (Denorm)	358 ± 113	301 ± 97	725 ± 158
G (Denorm+Idx)	430 ± 114	654 ± 226	1345 ± 363
p-value (ANOVA)	< 0.001	< 0.001	< 0.001

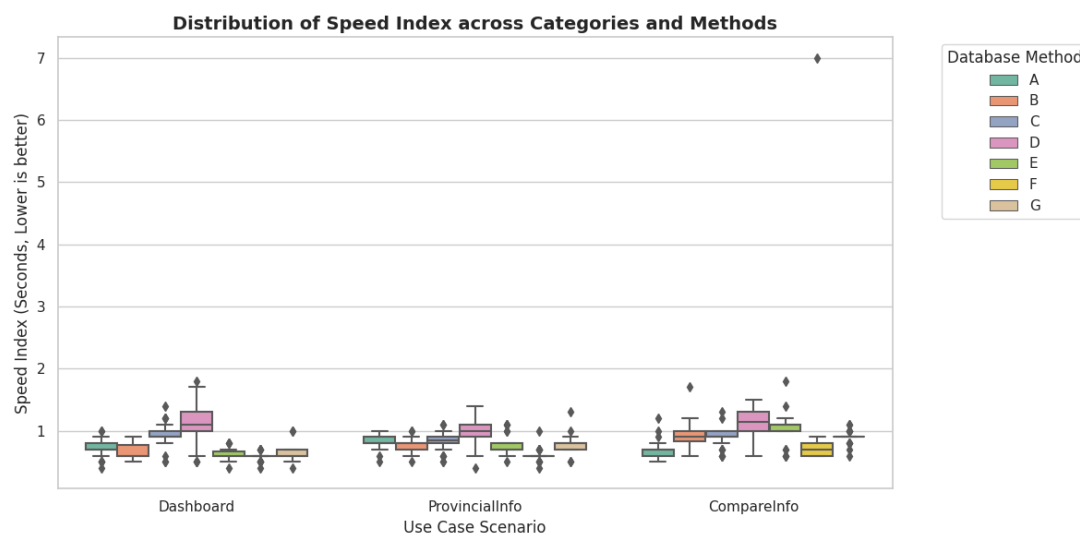


รูปที่ 5 Boxplot แสดงการกระจายของ TBT ในหน้า Dashboard แต่ละกรณี (Use Case Scenario)

การเปลี่ยนจากวิธี C มาเป็นวิธี F ช่วยลด TBT ได้อย่างมีนัยสำคัญ ($p < 0.001$) ในทุกสถานการณ์ นอกจากนี้ การเพิ่มดัชนี (Index) บนโครงสร้าง Normalization (วิธี B) ช่วยแก้ปัญหาคงความหน่วงของหน้า Dashboard ได้ดีกว่าวิธี C อย่างมีนัยสำคัญ ($p = 0.0002$)

ตารางที่ 4 เปรียบเทียบ Speed Index (SI) (หน่วย: วินาที, ค่ายิ่งต่ำแสดงถึงประสิทธิภาพที่ดีกว่า)

โครงสร้างฐานข้อมูล	กรณี Dashboard	กรณี ProvincialInfo	กรณี CompareInfo
A (Norm+Idx+Part)	0.72 ± 0.14	0.81 ± 0.10	0.70 ± 0.14
B (Norm+Idx)	0.67 ± 0.11	0.77 ± 0.12	0.92 ± 0.23
C (Norm, Baseline)	0.93 ± 0.19	0.83 ± 0.15	0.90 ± 0.16
D (Denorm+Idx+Part)	1.13 ± 0.32	0.99 ± 0.20	1.15 ± 0.27
E (Materialized View)	0.62 ± 0.09	0.75 ± 0.16	1.05 ± 0.24
F (Denorm)	0.59 ± 0.07	0.61 ± 0.10	0.92 ± 0.15
G (Denorm+Idx)	0.63 ± 0.12	0.78 ± 0.14	0.90 ± 0.10
p-value (ANOVA)	< 0.001	< 0.001	0.0172



รูปที่ 6 Boxplot แสดงการกระจายของ SI ในหน้า Dashboard แต่ละกรณี (Use Case Scenario)

ผลของ Speed Index สอดคล้องกับตัวชี้วัดอื่น โดยในหน้า Dashboard วิธี F และวิธี E สามารถทำความเร็ว Speed Index ได้ดีกว่าค่าตั้งต้นวิธี C อย่างมีนัยสำคัญทางสถิติ

จากการประเมินพบว่า ตัวชี้วัด Accessibility มีค่าคงที่ประมาณ 89 คะแนน Best Practices อยู่ในช่วง 98-99 คะแนน และ SEO มีค่า 100 คะแนนเต็ม ในทุกโครงสร้างฐานข้อมูล ปรากฏการณ์นี้

เกิดขึ้นเนื่องจากคะแนนกลุ่มนี้เป็นผลลัพธ์ที่ผูกติดกับการออกแบบโครงสร้างส่วนแสดงผล โครงสร้างแท็ก HTML และการตั้งค่าเซิร์ฟเวอร์เป็นหลัก ซึ่งได้รับการควบคุมให้เหมือนกันในการทดลองนี้ จึงมีความใกล้เคียงกันทุกวิธีและไม่สะท้อนผลกระทบจากการเปลี่ยนแปลงโครงสร้างฐานข้อมูล

6. อภิปรายผล

ผลการทดลองภาพรวมแสดงให้เห็นว่าโครงสร้างฐานข้อมูลแต่ละวิธีมีแนวโน้มแตกต่างกันและส่งผลกระทบต่อประสิทธิภาพหน้าเว็บอย่างมีนัยสำคัญทางสถิติ ($p < 0.05$) จากการทดสอบความแปรปรวนทางเดียว (One-way ANOVA) โดยคะแนนประสิทธิภาพ (Performance Score) ของทั้ง 7 วิธีอยู่ในช่วง 78.00 ถึง 84.13 ซึ่งสามารถนำมาอภิปรายผลเชิงวิศวกรรมข้อมูลและเปรียบเทียบกับงานวิจัยที่ผ่านมาตามองค์ประกอบสำคัญได้ดังนี้

6.1 ผลกระทบของการลดรูปบรรทัดฐานและเปรียบเทียบงานวิจัยก่อนหน้า

ผลการทดลองชี้ชัดว่า วิธี F การลดรูปบรรทัดฐานโดยไม่สร้างดัชนีและไม่แบ่งส่วนข้อมูลให้ประสิทธิภาพความเร็วและคะแนนรวมสูงสุด เหตุผลเชิงกลไกฐานข้อมูลมาจากการลดความจำเป็นในการเชื่อมโยงตาราง ซึ่งเป็นปัญหาคอขวดที่สำคัญในการประมวลผล สอดคล้องกับแนวคิดของ (Kimball & Ross 2013) ที่ระบุว่าการรวมข้อมูลไว้ในตารางข้อมูลจริง เพียงตารางเดียวช่วยให้การสืบค้นข้อมูลเชิงวิเคราะห์รวดเร็วขึ้น นอกจากนี้ การไม่สร้างดัชนียังช่วยลดความเสี่ยงภาระการประมวลผลส่วนเกินของระบบในการสแกนผ่านโครงสร้างแบบต้นไม้ (B-Tree) เมื่อต้องกวาดข้อมูลในปริมาณมาก

6.2 ข้อเสียที่ต้องแลกและค่าความผันผวนทางสถิติ

อย่างไรก็ตาม การใช้โครงสร้างแบบลดรูปบรรทัดฐาน (Denormalization) ย่อมแลกมาด้วยผลกระทบเชิงลบหรือข้อเสียที่ต้องแลก (Trade-off) ด้านพื้นที่จัดเก็บข้อมูลส่วนเกินที่เพิ่มขึ้นจากความซ้ำซ้อนของข้อมูล และความท้าทายในการรักษาความถูกต้องสมบูรณ์ของข้อมูล (Data integrity) ซึ่งจะแตกต่างจากโครงสร้างแบบทำบรรทัดฐาน (Normalization) ในวิธี C ที่สามารถรับประกันความถูกต้องของข้อมูลได้ดีกว่า (Tongrak, 2009) นอกจากนี้ จากผลการทดลองพบว่าค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) มีความผันผวนสูงในตัวชี้วัดระยะเวลาหยุดชะงักรวม (Total Blocking Time: TBT) ปรากฏการณ์นี้เกิดจากความผันผวนของการจัดสรรทรัพยากร ของระบบปฏิบัติการและเว็บเบราว์เซอร์ในการประมวลผลคำสั่งฝั่งผู้ใช้งาน (Client-side) ขณะรันทดสอบ ซึ่งเป็นพฤติกรรมปกติของการวัดผลในลักษณะนี้

6.3 ข้อจำกัดของการวิจัยและความสามารถในการนำไปใช้ทั่วไป

งานวิจัยนี้มีข้อจำกัดคือการทดสอบทำภายใต้ชุดข้อมูลระดับ 693,184 ระเบียบ และเป็นการจำลองพฤติกรรมการใช้งานแบบเน้นการอ่านข้อมูลเป็นหลัก (Read-heavy workload) ทั้งนี้ความสามารถในการนำผลไปใช้ทั่วไปจะมีความเหมาะสมและแม่นยำสูงกับระบบหน้าปัดแสดงผลข้อมูลทางสาธารณสุขที่มีโครงสร้างตัวแปรและปริมาณข้อมูลในระดับใกล้เคียงกัน

6.4 เงื่อนไขที่ผลอาจเปลี่ยนแปลง หากสภาพแวดล้อมถูกเปลี่ยนเงื่อนไขไปใช้กับชุดข้อมูลที่ขยายตัวอย่างก้าวกระโดดถึงระดับหลายสิบล้านระเบียบ หรือเป็นระบบที่มีการงานเน้นการเขียนและปรับปรุงข้อมูลตามเวลาจริง (Write-heavy workload) บ่อยครั้ง ผลลัพธ์อาจเปลี่ยนแปลงไป โดยโครงสร้างที่ไม่มีการสร้างดัชนีอย่างวิธี F อาจเกิดปัญหาคอขวดในการค้นหา และโครงสร้างที่มีดัชนีหนาแน่นอย่างวิธี G จะได้รับผลกระทบเชิงลบอย่างหนักจากภาระการประมวลผลส่วนเกินในการบันทึกข้อมูลดัชนีใหม่ทุกครั้งที่มีการปรับปรุงข้อมูล

6.5 นัยเชิงปฏิบัติ สำหรับนัยเชิงปฏิบัติในการนำไปใช้งานจริงผู้วิจัยขอเสนอแนะให้นักพัฒนาเลือกใช้วิธีการดังนี้

1. สำหรับระบบระยะเริ่มต้นหรือข้อมูลระดับหลักแสน ควรเลือกใช้วิธี F เนื่องจากพัฒนาระบบได้รวดเร็วและให้ความเร็วในการแสดงผลสูงสุด
2. สำหรับระบบที่เน้นหน้ากระดานสรุปข้อมูลภาพรวม ควรพิจารณาใช้วิธี E การทำ Materialized View เพื่อลดภาระการคำนวณซ้ำแบบทันทีทันใด
3. สำหรับเว็บไซต์ที่เน้นการเจาะลึกเงื่อนไขแบบเฉพาะเจาะจง ควรเลือกใช้วิธี G การลดรูปบรรทัดฐานร่วมกับการสร้างดัชนี เพื่อรักษาสมดุลระหว่างความเร็วในการสืบค้นข้อมูลและการรองรับการขยายตัวของข้อมูล

6.6 การประยุกต์ใช้ในหน่วยงานสาธารณสุขและการพัฒนาในอนาคต ผลลัพธ์เชิงประจักษ์จากการวิจัยครั้งนี้ สามารถนำไปประยุกต์ใช้ในหน่วยงานด้านสาธารณสุข เช่น กรมควบคุมโรค โรงพยาบาลส่วนภูมิภาค หรือศูนย์สารสนเทศระดับชาติ ในการตัดสินใจเลือกสถาปัตยกรรมข้อมูลเพื่อขับเคลื่อนนโยบาย ซึ่งสอดคล้องกับแนวคิดสถาปัตยกรรมคลังข้อมูลทางคลินิกระดับชาติของ Ahamed, (Khan, & Haque 2021) ที่ระบุว่าการประมวลผลข้อมูลสุขภาพขนาดใหญ่ต้องการโครงสร้างที่เอื้อต่อการสืบค้นที่รวดเร็วเพื่อตอบสนองต่อสถานการณ์ฉุกเฉิน ยิ่งไปกว่านั้น นัยเชิงวิศวกรรมข้อมูลของการทำ Denormalization (วิธี F และ G) ยังได้รับการสนับสนุนจากงานวิจัยของ (Mohsin 2025) ที่ชี้ว่าเทคนิคการออกแบบโครงสร้างตารางข้อมูลเพื่อหลีกเลี่ยงความซับซ้อนของคิวรี ส่งผลให้ระบบฐานข้อมูลเชิง

วิเคราะห์ทำงานได้เร็วขึ้นอย่างมีนัยสำคัญ สอดคล้องกับกลยุทธ์การปรับแต่งฐานข้อมูล (Database Optimization Strategies, 2023) และแนวทางการสร้างระบบประมวลผลประสิทธิภาพสูง (Building High-Performance OLTP Systems, 2025) ที่สรุปตรงกันว่า ภาระงานเชิงอ่านปริมาณมาก (Read-heavy workload) จะได้ประโยชน์สูงสุดจากการลดโครงสร้างตารางและการจัดการดัชนีที่เหมาะสม ซึ่งส่งผลโดยตรงต่อการจัดท่อส่งข้อมูลให้มีความกระชับ (Streamlined Data Pipeline) และเพิ่มขีดความสามารถให้แก่เว็บแดชบอร์ดในการดึงข้อมูลดิบปริมาณมหาศาลมาประมวลผลและนำเสนอแบบเวลาจริง (Real-time) ได้อย่างมีประสิทธิภาพสูงสุดในทางปฏิบัติ (Kukkadapu, 2025)

7. สรุปผลการวิจัย

งานวิจัยนี้ได้ทำการศึกษาและเปรียบเทียบประสิทธิภาพของโครงสร้างฐานข้อมูลทั้ง 7 วิธี สำหรับระบบหน้าปัดแสดงผลข้อมูล (Web dashboard) ใฝ่ระยะโรค ผลการศึกษพบว่าโครงสร้างฐานข้อมูลส่งผลต่อประสิทธิภาพความเร็วของหน้าเว็บอย่างมีนัยสำคัญทางสถิติ ($p < 0.05$) โดยวิธี F การลดรูปแบบพื้นฐานโดยไม่สร้างดัชนีและไม่แบ่งส่วนข้อมูล ให้ผลลัพธ์ประสิทธิภาพโดยรวมดีที่สุด แตกต่างจากโครงสร้างแบบดั้งเดิมอย่างมีนัยสำคัญทางสถิติ วิธีดังกล่าวให้คะแนนประสิทธิภาพ (Performance Score) สูงสุดเฉลี่ย 84.13 คะแนน รวมถึงมีดัชนีความเร็ว (Speed Index: SI) ต่ำที่สุดที่ 0.553 วินาที และมีระยะเวลาหยุดชะงักรวม (Total Blocking Time: TBT) ต่ำที่สุดที่ 395.33 มิลลิวินาที

เหตุผลเชิงกลไกด้านฐานข้อมูลที่ทำให้วิธี F มีประสิทธิภาพสูงสุด เป็นผลมาจากการออกแบบที่สามารถลดขั้นตอนการประมวลผลที่ต้องใช้การเชื่อมโยงตาราง หลายตารางซึ่งเป็นปัญหาคอขวดที่สำคัญ ประกอบกับการปราศจากภาระการประมวลผลส่วนเกินของระบบจัดการฐานข้อมูลที่ต้องสแกนค้นหาข้อมูลผ่านโครงสร้างดัชนี (Index) ส่งผลให้การอ่านข้อมูลในลักษณะการสแกนข้อมูลแบบลำดับจากตารางหลักเพียงตารางเดียวสามารถทำงานและคืนผลลัพธ์ได้รวดเร็วที่สุด

8. ข้อเสนอแนะ

1. ข้อเสนอแนะสำหรับการพัฒนาระบบ ควรเลือกโครงสร้างฐานข้อมูลให้สอดคล้องกับลักษณะภาระงาน (Workload) หากระบบเน้นการอ่านและสรุปผลข้อมูลจำนวนมาก ควรพิจารณาเลือกใช้วิธี F เป็นหลัก แต่หากข้อมูลในระบบมีการเขียนหรือปรับปรุงบ่อยครั้ง ควรพิจารณาประเด็นด้านความถูกต้องสมบูรณ์ของข้อมูลควบคู่ไปด้วย
2. ข้อเสนอแนะสำหรับงานวิจัยในอนาคต ควรศึกษาและเปรียบเทียบประสิทธิภาพเพิ่มเติมเมื่อปริมาณข้อมูลเพิ่มขึ้นสู่ระดับสิบล้านระเบียน เพื่อประเมินจุดอิ่มตัวของการรองรับการขยายตัวของแต่ละวิธี รวมถึงควรศึกษาผลกระทบของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูลกลุ่ม

NoSQL หรือฐานข้อมูลที่ออกแบบมาเพื่อระบบคลาวด์ (Cloud-native database) ซึ่งอาจมีความเหมาะสมกับโครงสร้างข้อมูลขนาดใหญ่ในอนาคต

3. ข้อเสนอแนะเชิงนโยบายและการนำผลการวิจัยไปใช้ประโยชน์ ผลการวิจัยนี้สามารถนำไปประยุกต์ใช้เป็น "พิมพ์เขียว (Blueprint)" สำหรับหน่วยงานสาธารณสุขระดับประเทศหรือระดับสากล เช่น กรมควบคุมโรค (Department of Disease Control) หรือ องค์การอนามัยโลก (WHO) ที่มีความจำเป็นต้องพัฒนาระบบเว็บแดชบอร์ดรายงานสถานการณ์โรคติดต่อ (เช่น ศูนย์ข้อมูล COVID-19 หรือระบบเฝ้าระวังโรคไข้หวัดใหญ่) หน่วยงานเหล่านี้สามารถนำแนวทางการลดรูปบรรทัดฐาน (Denormalization - วิธี F หรือ G) ไปปรับปรุงโครงสร้างฐานข้อมูลเชิงสัมพันธ์เดิมที่มีอยู่ ให้สามารถรองรับผู้ใช้งานพร้อมกันจำนวนมากและแสดงผลภาพรวมระดับประเทศได้อย่างรวดเร็ว โดยไม่จำเป็นต้องรื้อระบบเพื่อลงทุนในสถาปัตยกรรมข้อมูลใหม่ที่มีราคาสูง ทั้งนี้ เหตุผลสำคัญที่สนับสนุนคุณค่าของงานวิจัยนี้ คือในสถานการณ์วิกฤตการระบาด ความเร็วในการแสดงผลข้อมูลของหน้าปัด (Dashboard) ส่งผลโดยตรงต่อความรวดเร็วในการวิเคราะห์สถานการณ์ของผู้บริหาร การกระจายทรัพยากรทางการแพทย์ และการออกมาตรการควบคุมโรคที่ทันเวลาที่ การออกแบบฐานข้อมูลที่เหมาะสมจึงไม่ใช่เพียงการเพิ่มประสิทธิภาพทางเทคโนโลยีสารสนเทศ แต่เป็นการยกระดับขีดความสามารถในการตอบสนองต่อภาวะฉุกเฉินทางสาธารณสุขของประเทศอย่างเป็นรูปธรรม

9. กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จากหลายภาคส่วน ผู้วิจัยขอขอบพระคุณ อาจารย์ที่ปรึกษาและคณาจารย์ ที่ให้คำแนะนำทางวิชาการ ตรวจสอบความถูกต้อง และให้ข้อเสนอแนะเพื่อปรับปรุงงานวิจัยให้มีความสมบูรณ์ยิ่งขึ้น รวมถึงขอขอบคุณ หน่วยงานด้านสาธารณสุข ที่เผยแพร่ข้อมูลแบบเปิดเพื่อการศึกษา โดยเฉพาะ กรมควบคุมโรค (Department of Disease Control: DDC) และกองระบาดวิทยา สำหรับชุดข้อมูลผู้ป่วยโรคไข้หวัดใหญ่ที่นำมาใช้ในการทดลอง ซึ่งช่วยให้ผู้วิจัยสามารถดำเนินการวิเคราะห์และประเมินประสิทธิภาพของระบบเว็บแดชบอร์ดได้อย่างเป็นรูปธรรม

ท้ายที่สุด ผู้วิจัยขอขอบคุณทุกท่านที่มีส่วนเกี่ยวข้องในการสนับสนุนและให้กำลังใจตลอดกระบวนการดำเนินงานวิจัย ด้วยดี ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

10. เอกสารอ้างอิงอิเล็กทรอนิกส์

- กรมควบคุมโรค. (2567). รายงานสถานการณ์โรคไข้หวัดใหญ่. สืบค้นเมื่อ 14 เมษายน 2569, จาก <https://data.go.th/dataset/gdpublish-ddc-doe-011>
- ประกาศ ทองรัก. (2552). เครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ [วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ, จุฬาลงกรณ์มหาวิทยาลัย].
- พัชรียา ทองพูล, พิมพ์ชนก จำเริญ, รมย์นลิน บุญฤทธิ์, และ สายชล สิ้นสมบุญทอง. (2562). การเปรียบเทียบประสิทธิภาพในการทำนายผลการปรับความไม่สมดุลของข้อมูลในการจำแนกด้วยเทคนิคการทำเหมืองข้อมูล. วารสารวิทยาศาสตร์และเทคโนโลยีไทย, 8(6), 566-584.
- Ahamed, S. I., Khan, S. I., & Haque, A. S. L. (2021). A Privacy-Preserving National Clinical Data Warehouse: Architecture and Analysis. Retrieved from https://epublications.marquette.edu/cgi/viewcontent.cgi?article=1074&context=comp_fac
- Building High-Performance OLTP Systems: A Comprehensive Guide to Database Design and Implementation. (2025). *International Journal of Scientific and Academic Technologies (IJSAT)*, 16(1). <https://www.ijSAT.org/papers/2025/1/3007.pdf>
- Database Optimization Strategies: Enhancing Performance and Scalability. (2023). *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 12(11). from <https://ijcsmc.com/docs/papers/November2023/V12I11202313.pdf>
- Ghazisaeidi, M., Safdari, R., Torabi, M., Mirzaee, M., Farzi, J., & Goodini, A. (2015). Development of Performance Dashboards in Healthcare Sector: Key Practical Issues. *Acta Informatica Medica*, 23(5), 317-321. from <https://pmc.ncbi.nlm.nih.gov/articles/PMC4639357/>
- Kukkadapu, S. (2025). Real-Time Business Intelligence Dashboard For Financial Decision-Making: A Streamlined Data Pipeline Approach. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 12(4). From <https://www.jetir.org/papers/JETIR2504D06.pdf>

- Lakkimsetty. (2023). Database optimization strategies: Enhancing performance and scalability. *International Journal of Computer Science and Mobile Computing (IJCSMC)*,12(11).<https://ijcsmc.com/docs/papers/November2023/V12I11202313.pdf>
- Mohsin, M. (2025). Techniques for Logical Design and Efficient Querying of Data Warehouses. *International Journal of Novel Research and Development (IJNRD)*. from <https://www.ijnrd.org/papers/IJNRD2006037.pdf>