บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

ในปัจจุบันนี้ ระบบคลังข้อมูลเป็นสิ่งที่มีความสำคัญและจำเป็นมากขึ้นสำหรับองค์กรเพื่อนำมาใช้ในการสนับสนุนการตัดสินใจของผู้บริหาร โดยจะมีการดึงข้อมูลหรือสารสนเทศจากคลังข้อมูลขึ้นมาใช้งาน ซึ่งในการดึงข้อมูลขึ้นมาใช้งานแต่ละครั้ง ส่วนมากมักจะเป็นลักษณะของการสอบถามข้อมูลแบบ Ad-hoc และแบบซับซ้อน ทำให้การค้นหาข้อมูลก็ย่อมใช้เวลามากขึ้น วิธีการหนึ่งที่จะเพิ่มประสิทธิภาพการค้นหาข้อมูลให้รวดเร็วขึ้น โดยไม่ต้องเสียค่าใช้จ่ายเพิ่มนั้น คือ การทำดัชนี เพราะการทำดัชนีเราไม่ต้องเพิ่มอุปกรณ์ฮาร์ดแวร์ แต่เป็นการกรองข้อมูลให้เล็กลง ซึ่งการทำดัชนีนั้นก็มีหลายแบบด้วยกัน และแต่ละแบบก็เหมาะกับลักษณะข้อมูลที่แตกต่างกัน

ในงานวิทยานิพนธ์ชิ้นนี้ เป็นการพัฒนาขั้นตอนวิธี (Algorithm) เพื่อสร้างดัชนีสำหรับการค้นหาข้อมูลในคลังข้อมูล ซึ่งอยู่บนหลักการของการทำดัชนีแบบบิตแมป ที่เหมาะกับข้อมูลที่มีคาร์ดินอลิตี้ต่ำ มีค่าแตกต่างไม่มากนัก และเหมาะกับข้อมูลที่ไม่เปลี่ยนแปลง ซึ่งพบบ่อยในคลังข้อมูล ทั้งนี้เพื่อประหยัดพื้นที่ในการจัดเก็บดัชนี นอกจากนี้การทำดัชนีแบบบิตแมปยังมีคุณสมบัติของการดำเนินการระดับบิต (Bit Operation) ระหว่างบิตแมปเวกเตอร์ก่อนดึงข้อมูลจริง ทำให้ใช้เวลาน้อยลงในการค้นหาข้อมูล

ดัชนีที่อยู่บนพื้นฐานของบิตแมปเดิมที่มีอยู่ ได้แก่ ดัชนีบิตแมปแบบพื้นฐานแบบ Range แบบช่วง และแบบเข้ารหัส สำหรับดัชนีบิตแมปแบบใหม่ที่นำเสนอ เรียกว่า ดัชนีบิตแมปแบบกระจาย ดัชนีบิตแมปเหล่านี้ต่างก็มีคุณสมบัติ ข้อดี ข้อจำกัด แตกต่างกันไป ดังนั้นเราจึงควรเลือกประเภทของดัชนีแบบบิตแมปให้เหมาะสมกับการค้นหาข้อมูลบนคลังข้อมูล ซึ่งสังเกตได้จากความสัมพันธ์ระหว่างพื้นที่ในการจัดเก็บดัชนีและเวลาที่ใช้ในการค้นหาข้อมูล เพื่อให้เข้าใจได้ง่ายขึ้น จึงขอสรุปลำดับที่และขนาดความมากน้อยในการใช้พื้นที่ในการจัดเก็บดัชนีและเวลาที่ใช้ในการค้นหาข้อมูล ทั้งกรณีที่เป็นการสอบถามข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิก ได้ดังตาราง 6-1 และ 6-2

ตาราง 6-1 สรุปการใช้พื้นที่ในการจัดเก็บดัชนี

ลำดับที่	ชนิดของดัชนีบิตแมป	การใช้พื้นที่ในการจัดเก็บดัชนี
1	แบบเข้ารหัส	↑ น้อยที่สุด
2	แบบกระจาย	
3	แบบช่วง	
4	แบบพื้นฐาน	↓ มากที่สุด

ตาราง 6-2 สรุปเวลาที่ใช้ในการค้นหาข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิก

ลำดับที่	ชนิดของดัชนีบิตแมป	เวลาที่ใช้ในการค้นหาข้อมูล
1	แบบพื้นฐาน	↑ น้อยที่สุด
2	แบบกระจาย	
3	แบบช่วง	
4	แบบเข้ารหัส	↓ มากที่สุด

จากตาราง 6-1 และ 6-2 เป็นการสรุปการใช้พื้นที่ในการจัดเก็บดัชนีและสรุปเวลา จะเห็นว่าท่ามกลางดัชนีบนพื้นฐานของบิตแมปนั้น ดัชนีบิตแมปแบบกระจาย เป็นดัชนีบิตแมปที่เหมาะสมที่สุดสำหรับการสอบถามข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิก ทั้งนี้เพราะ

- ดัชนีบิตแมปแบบพื้นฐาน เหมาะสำหรับการสอบถามข้อมูลแบบค่าเท่ากันมากที่สุด เพราะจำนวนบิตแมปเวกเตอร์ที่ถูกอ่าน เท่ากับ 1 และไม่มีการดำเนินการตรรกะใด ๆ เกิดขึ้น รองลงมา คือ ดัชนีบิตแมปแบบกระจาย เพราะจำนวนบิตแมปเวกเตอร์ที่ถูกอ่าน เท่ากับ 2 และมีการดำเนินการตรรกะเกิดขึ้นเพียง 1 ตัวดำเนินการเท่านั้น คือ ตัวดำเนินการตรรกะ AND สำหรับดัชนีบิตแมปแบบเข้ารหัส จะใช้เวลามากที่สุดในการค้นหาข้อมูล
- ดัชนีบิตแมปแบบพื้นฐาน เหมาะสำหรับการค้นหาข้อมูลแบบความเป็นสมาชิกมากที่สุด เพราะสามารถดำเนินการตรรกะ OR ได้ทันที หลังจากค้นหาบิตแมปเวกเตอร์แต่ละค่า แต่ดัชนีบิตแมปแบบช่วงและแบบกระจายจะต้องเสียเวลาในการดำเนินการตรรกะก่อน

และดัชนีบิตแมปแบบเข้ารหัส จะต้องเสียเวลามากขึ้นในการอ่านค่าที่จะค้นหานั้นจากตารางการเทียบค่าก่อนว่ามีการเข้ารหัสบิตแมปเวกเตอร์ในรูปแบบอะไร แล้วไปอ่านบิตแมปเวกเตอร์ทั้ง $\lceil \log_2 C \rceil$ บิตแมปเวกเตอร์จากตารางดัชนี เพื่อดึงแต่ละค่าที่จะค้นหาก่อนดำเนินการตรรกะ OR ทำให้ดัชนีบิตแมปแบบพื้นฐานใช้เวลาน้อยที่สุดในการค้นหาข้อมูล และดัชนีบิตแมปแบบเข้ารหัสใช้เวลามากที่สุดในการค้นหาข้อมูล

แม้ว่า ดัชนีบิตแมปแบบพื้นฐาน มีประสิทธิภาพมากที่สุดในการค้นหาข้อมูลทั้งแบบค่าเท่ากันและแบบความเป็นสมาชิก เพราะใช้เวลาน้อยที่สุดในการค้นหาข้อมูล แต่ดัชนีบิตแมปแบบพื้นฐานจะใช้พื้นที่มากที่สุดในการจัดเก็บดัชนี (เท่ากับ CN บิต หรือ C บิตแมปเวกเตอร์) เนื่องจากมีการใช้ 1 บิตแมปเวกเตอร์ในการแทนค่าของแอทริบิวต์ 1 ค่า ซึ่งทำให้สิ้นเปลืองค่าใช้จ่ายในเรื่องพื้นที่จัดเก็บดัชนี โดยเฉพาะอย่างยิ่งในกรณีที่แอทริบิวต์ที่นำมาสร้างดัชนีมีค่าคาร์ดินอลิตี้สูง ๆ ทำให้ประสิทธิภาพลดลง ซึ่งเป็นข้อเสียของดัชนีบิตแมปแบบพื้นฐาน และถึงแม้ว่าดัชนีบิตแมปแบบช่วง จะใช้พื้นที่น้อยลง ($\left\lceil \frac{C}{2} \right\rceil N$ บิต หรือ $\left\lceil \frac{C}{2} \right\rceil$ บิตแมปเวกเตอร์) กล่าวคือ ดัชนีบิตแมปแบบช่วง ลดพื้นที่ลงครึ่งหนึ่งในการจัดเก็บดัชนี จากดัชนีบิตแมปแบบพื้นฐาน ซึ่งเป็นการแก้ปัญหาเรื่องขนาดพื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปแบบพื้นฐาน แต่ก็ยังจัดว่ามาก และดัชนีบิตแมปแบบเข้ารหัส แม้ว่าจะประหยัดพื้นที่ได้มากที่สุดในการจัดเก็บดัชนี (เท่ากับ $\lceil \log_2 C \rceil N$ บิต) แต่เมื่อพิจารณาค่าใช้จ่ายในเรื่องเวลาในการค้นหาข้อมูล จะเห็นได้ว่าใช้เวลามากที่สุดทั้งในการค้นหาแบบค่าเท่ากันและแบบความเป็นสมาชิก เพราะต้องอ่านบิตแมปเวกเตอร์ทั้ง $\lceil \log_2 C \rceil$ บิตแมปเวกเตอร์จากตารางดัชนีทุกครั้งที่มีการสอบถามข้อมูลและทุกค่าที่ต้องการค้นหา ซึ่งถ้าหากแอทริบิวต์ที่นำมาสร้างดัชนีมีค่าคาร์ดินอลิตี้สูง ๆ ก็จะทำให้เสียเวลาในการอ่านตารางเทียบค่าและตารางดัชนีทั้ง $\lceil \log_2 C \rceil$ บิตแมปเวกเตอร์

สำหรับดัชนีบิตแมปแบบกระจาย ซึ่งเป็นเทคนิคแบบใหม่ที่นำเสนอ ได้ขจัดปัญหาเหล่านี้ลง กล่าวคือ สามารถลดพื้นที่ในการจัดเก็บดัชนี (เท่ากับ $\lceil 2\sqrt{C} \rceil N$ บิต) ได้มากกว่าดัชนีบิตแมปแบบช่วง และใช้เวลาน้อยลงในการค้นหาข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิก โดยเฉพาะอย่างยิ่งในกรณีที่การค้นหาแบบความเป็นสมาชิก จำนวนค่าที่จะค้นหาเท่ากับ จำนวนสมาชิกในกลุ่ม Z (ค่า m) หรือ L และค่าที่ต้องการค้นหานั้นอยู่ภายในกลุ่ม Z หรือ L เดียวกัน

6.2 อุปสรรคและปัญหา

อุปสรรคและปัญหาที่พบจากการทำวิทยานิพนธ์ สรุปเป็นข้อ ๆ ได้ดังนี้

- การจัดเตรียมข้อมูล (โดยเฉพาะกรณีข้อมูลเยอะมากหลายล้านเรคอร์ด)
- การตรวจสอบความถูกต้องของการค้นหาข้อมูล

6.2.1 ปัญหาการจัดเตรียมข้อมูล

ในการจัดเตรียมข้อมูลเพื่อสร้างดัชนีบิตแมปแต่ละชนิดนั้น มีขั้นตอนในการจัดเตรียมหลายขั้นตอน (รายละเอียดอยู่ในภาคผนวก หัวข้อ ก.1) และข้อมูลทดสอบแต่ละตารางก็มีจำนวนเรคอร์ดที่แตกต่างกัน (รายละเอียดอยู่ในภาคผนวก หัวข้อ ก.2) ทำให้ค่อนข้างยุ่งยากในการจัดเตรียมและการประเมินเปรียบเทียบดัชนี ทั้งในประเด็นของเวลาและฮาร์ดแวร์ที่ใช้ในการรันผลเพื่อใช้ในการจัดเตรียม

ในกรณีที่ข้อมูลเยอะมาก ทำให้ไม่สามารถที่จะเปิดแฟ้มข้อมูลได้ จึงจำเป็นต้องตัดข้อมูลจากขนาดต่าง ๆ เช่น จากหลักล้านเรคอร์ด เหลือเป็น 200,000 เรคอร์ด

6.2.2 ปัญหาการตรวจสอบความถูกต้องของการค้นหาข้อมูล

การตรวจสอบความถูกต้องของการค้นหาข้อมูล ก็เป็นอีกปัญหาหนึ่ง ซึ่งอาจเกิดจากสาเหตุต่าง ๆ ดังนี้

- การจัดเตรียม สาเหตุนี้จะยากต่อการตรวจสอบว่าความผิดพลาดจากการเตรียมข้อมูลขั้นตอนใด เช่น อาจเกิดจากการถ่ายโอนแฟ้มข้อมูลผ่านระบบเครือข่าย การดึงเฉพาะหมายเลขเรคอร์ด การเรียงลำดับหมายเลขเรคอร์ด การจัดเก็บข้อมูลในรูปแบบบิต
- การค้นหาข้อมูล ในการตรวจสอบความถูกต้องของการค้นหาข้อมูลว่าถูกต้องครบถ้วนหรือไม่นั้น จะค่อนข้างยุ่งยาก เนื่องจากจะต้องตรวจสอบความถูกต้องครบถ้วนด้วยมือและด้วยเครื่องมือ ซึ่งการตรวจสอบด้วยมือนั้นโดยการให้โปรแกรมแสดงผลบนจอภาพแล้วสังเกตดูว่าถูกต้องครบถ้วนหรือไม่ และการตรวจสอบด้วยเครื่องมือนั้น โดยการจัดเก็บผลจากการค้นหาข้อมูลนั้นเป็นแฟ้มข้อมูล แล้วนำแฟ้มข้อมูลที่ได้จากการค้นหา และแฟ้มข้อมูลเดิมมาเปรียบเทียบความแตกต่างกันด้วยคำสั่ง diff (รายละเอียดอยู่ในภาคผนวก หัวข้อ ข.2) ในระบบปฏิบัติการ Linux

6.3 ข้อเสนอแนะและงานในอนาคต

ดัชนีบิตแมปแบบกระจายมีข้อดีในเรื่องของการประหยัดพื้นที่ในการจัดเก็บดัชนีและประหยัดเวลาในการค้นหาข้อมูล กล่าวคือ เหมาะสำหรับการสอบถามข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิก โดยเฉพาะอย่างยิ่งจำนวนค่าที่ต้องการค้นหา เท่ากับ จำนวนสมาชิกในกลุ่ม Z (ค่า m) หรือ L และค่าที่ต้องการค้นหานั้นอยู่ภายในกลุ่ม Z หรือ L เดียวกัน

สำหรับข้อเสนอแนะต่าง ๆ และงานวิจัยที่น่าสนใจที่ควรจะทำในอนาคต มีดังนี้

- เทคนิคการทำเหมืองข้อมูล อาจเป็นทางเลือกหนึ่งในการช่วยหาความสัมพันธ์ของข้อมูลและการจัดกลุ่ม เพื่อให้กลุ่มข้อมูลที่จะดึงขึ้นมาใช้งานพร้อมกันให้อยู่ในกลุ่ม Z หรือ L เดียวกัน ซึ่งเป็นการเพิ่มประสิทธิภาพการค้นหาข้อมูลแบบความเป็นสมาชิก

เพราะดัชนีบิตแมปแบบกระจายจะดึงเพียงแค่ 1 บิตแมปเวกเตอร์ ในการตอบคำถาม ถ้าข้อมูลอยู่ในกลุ่มเดียวกัน

- ถ้าหากเราสามารถที่จะหาวิธีในการตรวจสอบว่าค่าที่ต้องการค้นหาแบบความเป็นสมาชิกอยู่ในกลุ่ม Z หรือ L เดียวกันหรือไม่ ได้อย่างมีประสิทธิภาพ (รวดเร็ว) ขึ้น เช่น ใช้วิธีการแฮช ก็จะทำให้ดัชนีบิตแมปแบบกระจายใช้เวลาน้อยลงกว่าเดิมมากในการค้นหาข้อมูลแบบความเป็นสมาชิกที่อยู่ในกลุ่ม Z หรือ L เดียวกัน และใช้เวลาน้อยกว่าดัชนีบิตแมปแบบพื้นฐานด้วย เพราะในกรณีที่เป็นการสอบถามข้อมูลแบบความเป็นสมาชิกที่จำนวนค่าที่ต้องการค้นหาเท่ากับจำนวนสมาชิกในกลุ่ม Z หรือ L และค่าที่ต้องการค้นหาอยู่ในกลุ่ม Z หรือ L เดียวกัน ดัชนีบิตแมปแบบกระจายสามารถดึงข้อมูลเหล่านั้นได้โดยการอ่านเพียงบิตแมปเวกเตอร์ Z หรือ L เพียง 1 บิตแมปเวกเตอร์เท่านั้นในการตอบคำถาม
- งานวิจัยที่น่าสนใจที่ควรจะทำในอนาคต คือ เทคนิคการค้นหาข้อมูลแบบช่วงให้มีประสิทธิภาพของดัชนีบิตแมปแบบกระจาย