การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

Scatter Bitmap Index Optimization using Data Clustering

แวฮาซัน แวหะมะ

Weahason Weahama

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาการจัดการเทคโนโลยีสารสนเทศ

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Management of Information Technology

Prince of Songkla University

2552

ชื่อวิทยานิพนธ์ การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ผู้เขียน นายแวฮาซัน แวหะมะ

สาขาวิชา การจัดการเทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก **คณะกรรมการสอบ**

....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล) (ผู้ช่วยศาสตราจารย์ ดร.โอม ศรนิล)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

.....กรรมการ

(ดร.ลัดดา ปรีชาวีรกุล)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาการจัดการเทคโนโลยีสารสนเทศ

.....

(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ผู้เขียน นายแวฮาซัน แวหะมะ

สาขาวิชา การจัดการเทคโนโลยีสารสนเทศ

ปีการศึกษา 2551

บทคัดย่อ

การค้นหาข้อมูลหรือสารสนเทศในคลังข้อมูลเพื่อช่วยในการตัดสินใจ ซึ่งส่วนมากจะเป็นลักษณะการค้นหาข้อมูลแบบซับซ้อนและแบบทันทีทันใด การทำดัชนีบิตแมปเป็นเทคนิคหนึ่งที่ใช้ในการเพิ่มประสิทธิภาพการค้นหาข้อมูล ดัชนีบิตแมปแบบกระจายใช้พื้นที่ในการจัดเก็บข้อมูลน้อย และยังมีคุณสมบัติของการดำเนินการระดับบิต ระหว่างบิตแมปเวกเตอร์ก่อนดึงข้อมูลจริง ทำให้ใช้เวลาน้อยลงในการค้นหาข้อมูล

วิทยานิพนธ์นี้ เป็นการประยุกต์วิธีการ K-mode clustering ในการแบ่งกลุ่มข้อมูลของดัชนีบิตแมปแบบกระจาย เพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูล จากผลการวิเคราะห์และการทดลองเปรียบเทียบระหว่างดัชนีบิตแมปแบบกระจายและดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล พบว่าดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลมีประสิทธิภาพในแง่ Space-Time Trade-off (การแลกเปลี่ยนระหว่างประสิทธิภาพของพื้นที่กับเวลา) สำหรับการสอบถามแบบสมาชิกมากกว่าดัชนีบิตแมปแบบกระจายที่เคยมีมา

Thesis Title	Scatter Bitmap Index Optimization using Data Clustering
Author	Mr. Weahason Weahama
Major Program	Management of Information Technology
Academic Year	2008

ABSTRACT

Decision support systems that retrieve information from a data warehouse environment are usually designed to process complex and ad hoc queries. Indexing techniques based on bitmap representations can be used to improve the efficiency of information retrieval. Scatter Bitmap Index uses less space and is more CPU-efficient than other bitmap indexing techniques. It is simple to represent, and improves query processing time by utilizing low-cost Boolean operations and multiple index scans. The Scatter Bitmap Index technique performs simple predicate conditions on the index level before going to the primary data source.

In this thesis, K-mode clustering is applied to group indexed attribute values. This reduces the number of bitmap vectors to be accessed. The comparative study and experimental results show that, in the best case, Scatter Bitmap Index Optimization using Data Clustering performs better than existing techniques for membership queries, in terms of space-time trade-off.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

รายการตาราง

รายการตาราง(ต่อ)

รายการภาพประกอบ

รายการภาพประกอบ(ต่อ)

รายการภาพประกอบ(ต่อ)

บทที่ 1

บทนำ

1.1 ความเป็นมา

คลังข้อมูล เป็นฐานข้อมูลที่จัดเก็บข้อมูลทั้งในอดีตและปัจจุบัน ซึ่งเก็บรวบรวมข้อมูลจากฐานข้อมูลประจำวัน และแหล่งข้อมูลจากภายนอก ข้อมูลอาจนำมาจากส่วนต่างๆ ขององค์กร ซึ่งจะถูกเปลี่ยนให้อยู่ในรูปแบบที่มีมาตรฐานเดียวกันทั้งหมด เพื่อให้สามารถนำไปใช้ในการวิเคราะห์ หรือ ช่วยการตัดสินใจให้ผู้บริหารได้ การดึงข้อมูลจากคลังข้อมูลขึ้นมาใช้งานแต่ละครั้ง ส่วนมากมักจะเป็นลักษณะของการสอบถามข้อมูลแบบซับซ้อน (Complex Query) [13] กล่าวคือ จะไม่ทราบว่าผู้ใช้จะสอบถามข้อมูลอะไรบ้าง จึงต้องมีการรวบรวมข้อมูลจากหลายแหล่งมาไว้ที่ศูนย์กลาง และเก็บบันทึกทั้งข้อมูลในอดีตและปัจจุบันไว้ในคลังข้อมูล ขอบเขตของข้อมูลที่เก็บไว้จึงกว้างขวาง ครอบคลุมทุกแง่มุมที่สำคัญขององค์กรนั้น ๆ เพื่อให้พร้อมใช้งานในการสอบถามข้อมูลทุกรูปแบบ ดังนั้นคลังข้อมูลจะมีขนาดใหญ่ ทำให้การค้นหาข้อมูลย่อมใช้เวลามากขึ้น

วิธีการหนึ่ง ในการลดเวลาในการค้นหาข้อมูล (เพิ่มประสิทธิภาพ) โดยไม่ต้องเสียค่าใช้จ่ายในการเพิ่มอุปกรณ์ฮาร์ดแวร์ คือ การทำดัชนีแบบบิตแมป [1,2,3,4,9,11,12,14,16] โดยดัชนีแบบบิตแมป เหมาะกับข้อมูลที่แอทริบิวต์มีค่าคาร์ดินอลิตี้ต่ำ (จำนวนค่าที่แตกต่างกันบนคอลัมน์ที่นำมาทำดัชนีมีค่าแตกต่างไม่มากนัก) เช่น เพศ จังหวัด เป็นต้น และเหมาะกับข้อมูลที่ไม่เปลี่ยนแปลง ซึ่งพบบ่อยในคลังข้อมูล [2,3,4,5,6,7,8,11,14,17]

ได้มีงานวิจัยที่เกี่ยวกับการทำดัชนีแบบบิตแมป ในคลังข้อมูลมากมาย ซึ่งงานวิจัยแต่ละชิ้นมุ่งเน้นในการลดพื้นที่ในการจัดเก็บ และเพิ่มความเร็วในการค้นหาข้อมูล หรือทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล และได้มีงานวิจัยที่ยังแยกประเภทของการค้นหาข้อมูล (Query) เข้ามาเกี่ยวข้อง ซึ่งแยกเป็นการค้นหาแบบเท่ากันและค้นหาแบบสมาชิก เช่น ดัชนีบิตแมปแบบพื้นฐาน [2] เป็นดัชนีที่มีความเร็วในการค้นหาข้อมูลแบบเท่ากับ แต่ไม่เหมาะสำหรับแอทริบิวต์สูงๆ เพราะใช้พื้นที่จัดเก็บมาก ดัชนีบิตแมปแบบเข้ารหัส [4] ซึ่งเป็นดัชนีบิตแมปที่เน้นการลดพื้นที่การจัดเก็บแต่ใช้เวลานานในการค้นหาข้อมูล ดัชนีบิตแมปแบบกระจาย [3,11] จะรวมทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล เป็นต้น นอกจากนี้เรายังพบอีกว่าจากโครงสร้างของดัชนีบิตแมปแบบกระจาย [3] เรายังสามารถเพิ่มความเร็วในการค้นหาข้อมูลได้อีก โดยการพิจารณาการค้นหาข้อมูลที่อยู่ในกลุ่มสมาชิกเดียวกัน

งานวิจัยนี้ ได้เสนอเทคนิคการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้น โดยใช้การแบ่งกลุ่มข้อมูล (Data Clustering) [5,6,7,9,10,17] ซึ่งเป็นหนึ่งในเทคนิคการทำเหมืองข้อมูล (Data Mining) [5,6,7,10,11,13,17]

มาช่วยในการจัดกลุ่มและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจาย [3,11]

1.2 การตรวจเอกสารและงานวิจัยที่เกี่ยวข้อง

1.2.1 การตรวจเอกสาร

ในระบบคลังข้อมูล การทำดัชนีแบบบิตแมปเป็นเทคนิคหนึ่งที่จะช่วยให้การสอบถามข้อมูลมีประสิทธิภาพโดยไม่ต้องเสียค่าใช้จ่ายในการเพิ่มฮาร์ดแวร์ใด ๆ ทั้งสิ้นซึ่งจะช่วยให้การประมวลผลมีความเร็วมากยิ่งขึ้น เนื่องจากสามารถดำเนินการระดับบิตระหว่างบิตแมปเวกเตอร์ก่อนดึงข้อมูลจริง การจัดการข้อมูลบนคลังข้อมูลนั้นจะต้องมีประสิทธิภาพทั้งในเรื่องของการจัดเก็บและการเข้าถึงข้อมูล โดยเฉพาะอย่างยิ่งการเข้าถึงข้อมูลจะต้องเป็นไปอย่างรวดเร็ว เนื่องจากการสอบถามมักเป็นแบบทันทีทันใด คือ ไม่ทราบล่วงหน้าว่าผู้ใช้จะสอบถามอะไร ในงานวิจัยที่ผ่านมามีการคิดค้นเทคนิคการทำดัชนีบิตแมปขึ้นมาหลายวิธี ซึ่งงานวิจัยแต่ละชิ้นมุ่งเน้นในการลดพื้นที่ในการจัดเก็บ และเพิ่มความเร็วในการค้นหาข้อมูล หรือ ทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล และได้มีงานวิจัยที่ยังแยกประเภทของการค้นหาข้อมูล (Query) เข้ามาเกี่ยวข้อง โดยมีการพัฒนาประสิทธิภาพทั้งในด้านการใช้พื้นที่และเวลาให้ดีขึ้นเรื่อย ๆ

1.2.2 งานวิจัยที่เกี่ยวข้อง

An Overview of Data Warehousing and OLAP Technology

งานวิจัยนี้ [19] กล่าวถึงสถาปัตยกรรมของเทคโนโลยีคลังข้อมูล การออกแบบเชิงแนวคิด ความสำคัญของเทคโนโลยีคลังข้อมูล ความแตกต่างระหว่างฐานข้อมูลดำเนินการและคลังข้อมูล เครื่องมือในการสร้างคลังข้อมูล เครื่องมือในการสอบถามและวิเคราะห์ข้อมูล วิธีการออกแบบคลังข้อมูล การเพิ่มประสิทธิภาพในการสอบถาม การจัดการคลังข้อมูล งานวิจัยในคลังข้อมูล

An Efficient Bitmap Encoding Scheme for Selection Queries

งานวิจัยนี้ [2] กล่าวถึงลักษณะการสอบถามข้อมูลในคลังข้อมูลชนิดต่าง ๆ การใช้ดัชนีบิตแมปเพื่อเพิ่มประสิทธิภาพการสอบถาม การทำดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index) ซึ่งมีการสอบถามที่รวดเร็ว มีการนำเสนอเทคนิคการทำดัชนีบิตแมปแบบช่วง(Interval Bitmap Index) ซึ่งใช้พื้นที่ในการจัดเก็บดัชนีบิตแมปเป็นครึ่งหนึ่งของดัชนีบิตแมปแบบพื้นฐาน แต่ต้องใช้เวลาในการสอบถามมากขึ้น

Encoded Bitmap Indexing for Data Warehouses

งานวิจัยนี้ [4] ได้กล่าวถึงปัญหาที่สำคัญในระบบคลังข้อมูล โดยเฉพาะพื้นที่ในการจัดเก็บดัชนี ดัชนีบิตแมปต่าง ๆจะมีการใช้พื้นที่ในการจัดเก็บเป็นจำนวนมาก จึงได้มีการนำเสนอเทคนิคการทำดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index) ซึ่งใช้พื้นที่ในการจัดเก็บดัชนีน้อยกว่าดัชนีบิตแมปชนิดอื่น แต่ดัชนีบิตแมปแบบเข้ารหัสต้องใช้เวลาในการสอบถามมากกว่าดัชนีบิตแมปชนิดอื่น ๆ

Scatter Bitmap : Space-Time Efficient Bitmap Indexing for Equality and Membership Queries

งานวิจัยนี้ [3] กล่าวถึงการทำดัชนีบิตแมปบนคลังข้อมูลในแบบต่าง ๆ การทำดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index) การทำดัชนีบิตแมปแบบช่วง (Interval Bitmap Index) การทำดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index) และได้นำเสนอเทคนิคการทำดัชนีบิตแมปแบบกระจาย (Scatter Bitmap Index) ซึ่งใช้พื้นที่ในการทำดัชนีบิตแมปน้อยกว่าการทำดัชนีบิตแมปแบบช่วง ในขณะที่ยังคงประสิทธิภาพในการสอบถามแบบค่าเท่ากันและแบบความเป็นสมาชิกอยู่

1.3 วัตถุประสงค์

เพื่อทำให้การค้นหาแบบความเป็นสมาชิกของดัชนีบิตแมปแบบกระจายมีประสิทธิภาพในการค้นหาข้อมูลยิ่งขึ้น โดยเฉพาะอย่างยิ่งข้อมูลที่ต้องการค้นหานั้นอยู่ในกลุ่มเดียวกัน

1.4 วิธีการดำเนินการวิจัย

1. ศึกษาแนวคิดที่เกี่ยวข้องกับระบบคลังข้อมูล และดัชนีบิตแมปทั้ง 4 ชนิด ได้แก่ ดัชนีบิตแมปแบบพื้นฐาน ดัชนีบิตแมปแบบช่วง ดัชนีบิตแมปแบบกระจาย และดัชนีบิตแมปแบบเข้ารหัส

2. วิเคราะห์ พัฒนาและออกแบบดัชนีบิตแมปแบบกระจายวิธีใหม่เพื่อลดเวลาการค้นหาข้อมูลโดยเฉพาะอย่างยิ่งข้อมูลที่ต้องการค้นหานั้นอยู่ในกลุ่มเดียวกัน

3. กำหนดรูปแบบการประเมินดัชนีบิตแมปแบบกระจายวิธีใหม่ และดัชนีบิตแมปแบบกระจายทั่วไปดังขั้นตอนต่อไปนี้

3.1 ทำการศึกษาข้อมูลสำหรับใช้ในการทดสอบ ซึ่งเป็นข้อมูลมาตรฐานจาก TPC-H Benchmark [18]

3.2 ติดตั้งโปรแกรมสำหรับรันผลการสร้างข้อมูลทดสอบจากการวัดเปรียบเทียบสมรรถนะ TPC-H บนระบบปฏิบัติการ Fedora Core 6 โดยจัดเก็บฐานข้อมูลทดสอบในรูปของ Flat File

3.3 จัดเตรียมข้อมูลทดสอบ โดยการเลือกเฉพาะแอทริบิวต์ที่สนใจนำมาทำดัชนี และ ปรับแต่งค่าข้อมูล (ดูภาคผนวก ก)

3.4 ออกแบบขั้นตอนวิธีในการสร้างดัชนีบิตแมปทั้ง 2 ชนิด คือ ดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (ดูภาคผนวก ก)

3.5 พัฒนาโปรแกรมเพื่อสร้างดัชนีบิตแมปทั้ง 2 ชนิด ตามขั้นตอนที่ออกแบบไว้ตามข้อ 3.4 โดยใช้ตัวแปลภาษาซี

4. ออกแบบขั้นตอนวิธีการสอบถามแบบต่าง ๆของดัชนีบิตแมปทั้ง 2 ชนิด คือ ดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (ดูภาคผนวก ก)

5. พัฒนาโปรแกรมเพื่อสอบถามแบบต่าง ๆบนดัชนีบิตแมปทั้ง 2 ชนิด ตามขั้นตอนที่ออกแบบไว้ตามข้อ 4 โดยใช้โปรแกรมภาษาซี

6. ประเมินประสิทธิภาพการใช้พื้นที่ในการจัดเก็บดัชนี (Space) ของดัชนีบิตแมปทั้ง 2 ชนิด

7. ประเมินประสิทธิภาพการใช้เวลาในการสอบถามแบบต่าง ๆ (Time) บนดัชนีบิตแมปทั้ง 2 ชนิด ด้วยการรันโปรแกรมที่พัฒนาขึ้นตามข้อ 5 แล้วบันทึกเวลาในการสอบถามของดัชนีบิตแมปแต่ละชนิด

8. ตรวจสอบความถูกต้องของการค้นหาข้อมูล (ดูภาคผนวก ข)

9. วิเคราะห์และสรุปผลการประเมินประสิทธิภาพดัชนีบิตแมปทั้ง 2 ชนิด

1.5 ขอบเขตงานวิจัย

1. วิเคราะห์ พัฒนาและออกแบบดัชนีบิตแมปแบบกระจายวิธีใหม่เพื่อลดเวลาในการค้นหาข้อมูลโดยเฉพาะอย่างยิ่งข้อมูลที่ต้องการค้นหานั้นอยู่ในกลุ่มเดียวกัน

2. ประเมินประสิทธิภาพของดัชนีบิตแมปแบบกระจายวิธีใหม่สำหรับการค้นหาข้อมูลในคลังข้อมูลเปรียบเทียบกับดัชนีบิตแมปแบบกระจายทั่วไป

1.6 ขั้นตอนการดำเนินงาน

1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้องและกำหนดขอบเขตของปัญหาให้ชัดเจน

2. วิเคราะห์ พัฒนาและออกแบบดัชนีบิตแมปแบบกระจายวิธีใหม่เพื่อลดเวลาการค้นหาข้อมูลโดยเฉพาะอย่างยิ่งข้อมูลที่ต้องการค้นหานั้นอยู่ในกลุ่มเดียวกัน

3. กำหนดรูปแบบการประเมินดัชนีบิตแมปแบบกระจายวิธีใหม่และดัชนีบิตแมปแบบกระจายทั่วไป

4. ศึกษาและวิเคราะห์หาเครื่องมือสำหรับใช้ประเมินดัชนีบิตแมปแบบกระจายวิธีใหม่และดัชนีบิตแมปแบบกระจายทั่วไป

5. พัฒนาดัชนีการค้นหาข้อมูล ตามที่ได้ทำการออกแบบไว้

6. ดำเนินการประเมินดัชนีบิตแมปแบบกระจายวิธีใหม่กับดัชนีบิตแมปแบบกระจายทั่วไป

7. สรุปผลจากการพัฒนา การทดสอบ และการประเมินเปรียบเทียบดัชนีทั้ง 2 ชนิด คือ ดัชนีบิตแมปแบบกระจายวิธีใหม่กับดัชนีบิตแมปแบบกระจายทั่วไป

8. จัดทำเอกสารประกอบการวิจัย

1.7 ระยะเวลาดำเนินงานและแผนการดำเนินงาน

ระยะเวลาดำเนินงาน มีนาคม พ.ศ. 2551 – มีนาคม พ.ศ. 2552

ตาราง 1-1 แสดงระยะเวลาดำเนินงาน

ขั้นตอนที่	เดือน												
	2551										2552		
	3	4	5	6	7	8	9	10	11	12	1	2	3
1	←	→											
2		←	→										
3			←	→									
4				←	→								
5						←	→						
6									←	→			
7										←	→		
8					←	→							

1.8 เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ สำหรับที่ใช้ในการเตรียมข้อมูลจาก TPC-H และพัฒนาโปรแกรมเพื่อประเมินประสิทธิภาพของเวลาที่ใช้ในการสอบถามแบบต่าง ๆบนดัชนีบิตแมปทั้ง 2 ชนิด ซึ่งมีสมรรถนะดังนี้

- ฮาร์ดแวร์จำนวน 2 เครื่อง

เครื่องคอมพิวเตอร์สำหรับเตรียมข้อมูลจากการวัดเปรียบเทียบสมรรถนะ TPC-H มีสมรรถนะดังนี้

CPU : Pentium Celeron 1.69 GHz

HDD : 40 GB.

RAM : 256 MB.

เครื่องคอมพิวเตอร์ สำหรับพัฒนาและประเมินประสิทธิภาพของดัชนีบิตแมป มีสมรรถนะดังนี้

CPU : Pentium Celeron 1.69 GHz

HDD : 40 GB.

RAM : 256 MB.

- ซอฟต์แวร์

- ระบบปฏิบัติการ : Linux Fedora core 6 สำหรับเตรียมข้อมูลจากการวัดเปรียบเทียบสมรรถนะ TPC-H
- gcc-4.1.2-13.fc6 สำหรับการทำงานคู่กับภาษา C
- samba-3.0.24-7.fc6 สำหรับการจัดการ file
- SSH Secure Shell Client และ putty สำหรับการเข้าถึง server
- ระบบปฏิบัติการ Microsoft Windows XP สำหรับพัฒนาและประเมินประสิทธิภาพของดัชนีบิตแมป

1.9 ประโยชน์ที่คาดว่าจะได้รับ

ได้ดัชนีบิตแมปแบบกระจายวิธีใหม่ที่มีประสิทธิภาพเพิ่มขึ้น ในแง่ของการลดเวลาการสอบถามข้อมูลแบบสมาชิก

บทที่ 2

ฐานข้อมูล, คลังข้อมูล, การค้นหาข้อมูล, การทำเหมืองข้อมูล และ การแบ่งกลุ่มข้อมูล

สำหรับบทนี้ กล่าวถึงคุณสมบัติของฐานข้อมูล สถาปัตยกรรมคลังข้อมูล วิธีการค้นหาข้อมูล ประโยชน์และเทคนิคต่าง ๆของการทำเหมืองข้อมูล ขั้นตอนเบื้องต้นของการวิเคราะห์ข้อมูล โดยจะแบ่งชุดข้อมูลออกเป็นกลุ่ม

ดัชนี (index) [1,2,3,4,9,11,12,14,16] มีความสำคัญมากต่อฐานข้อมูล เนื่องจาก ระบบฐานข้อมูลจะใช้ดัชนีในการค้นหาเรคอร์ดที่ต้องการได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้นจะถูกจัดเก็บไว้แยกจากตารางในพื้นที่ฐานข้อมูล โดยปกติถ้าไม่มีการประกาศดัชนีไว้ การค้นหาข้อมูลในตารางนั้นจะต้องทำการอ่านข้อมูลแบบเรียงลำดับ จากแถวที่หนึ่ง จนถึงแถวสุดท้าย ทำให้ใช้เวลานานในการค้นหาข้อมูล การสร้างดัชนีสำหรับตารางใด ๆ จะทำได้โดยการเลือกแอทริบิวต์ใดแอทริบิวต์หนึ่ง หรือหลายแอทริบิวต์จากตารางข้อมูลมาสร้างเป็นดัชนี และตารางหนึ่ง ๆ สามารถมีได้หลายดัชนี ดัชนีมีหลายแบบด้วยกัน เช่น ดัชนีแบบบิตแมป [1,2,3] ดัชนีแบบ B-Tree [1,2,3] เป็นต้น ซึ่งดัชนีแต่ละแบบจะเหมาะกับข้อมูลแตกต่างกัน โดยดัชนีแบบบิตแมปเหมาะกับข้อมูลที่มีคาร์ดินอลิตี้ต่ำ ส่วนดัชนีแบบ B-Tree เป็นดัชนีที่นิยมใช้กับข้อมูลที่มีคาร์ดินอลิตี้สูง

การพิจารณาเลือกแอทริบิวต์มาทำเป็นดัชนี จะทำการเลือกแอทริบิวต์ที่อยู่หลัง WHERE clause ที่มีการดำเนินการ {=, <, >, <=, >=, BETWEEN, IN} แอทริบิวต์ใน GROUP BY clause และ ORDER BY clause นอกจากการเลือกดัชนีที่เหมาะสมแล้ว ยังต้องพิจารณาค่าใช้จ่ายในการทำดัชนี เช่น ค่าใช้จ่ายในการใช้พื้นที่ในการจัดเก็บดัชนี เป็นต้น งานวิทยานิพนธ์จะมีทฤษฎีที่เกี่ยวข้องดังนี้

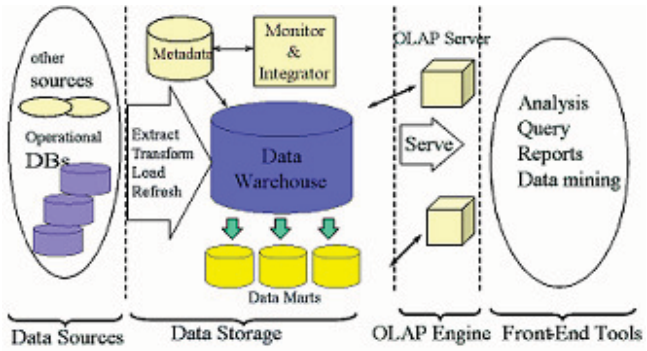
2.1 ฐานข้อมูล (Database)

ปัจจุบันมีการใช้ฐานข้อมูลในระบบงานทั่วไป (Operational Database) [1,12,15] ที่นิยมใช้อยู่ในปัจจุบันมีหลักในการเก็บข้อมูลที่เน้นในเรื่อง ลดการสูญหายของข้อมูล (Information Lost) รักษาความถูกต้อง (Integrity) การลดความซ้ำซ้อน (Redundancy) และลดความผิดพลาดที่เกิดขึ้นจากการแก้ไขข้อมูล (Update Anomalies) จึงมีความสามารถเพียงแค่การเรียกใช้ข้อมูลที่มีอยู่ แต่ไม่สามารถจะนำมาช่วยในการสนับสนุนการตัดสินใจได้ เพราะเมื่อมีการเรียกใช้ข้อมูลจะต้องเรียกใช้ข้อมูลจากฐานข้อมูลขนาดใหญ่ ซึ่งมีข้อมูลจำนวนมากและมีการแตกตารางที่นอร์มัลไลซ์ (Normalized Table) แล้วออกเป็นหลายตาราง จึงไม่รองรับคำถามที่ต้องการ

จะนำมาใช้ช่วยในการสนับสนุนการตัดสินใจ (Decision Support Queries) ที่มีการรวม (Join) กันของตารางต่าง ๆที่ซับซ้อน ซึ่งจะทำให้ประสิทธิภาพของการค้นหาข้อมูลจากฐานข้อมูลลดลง นอกจากนี้การเก็บข้อมูลในระบบฐานข้อมูลทั่วไป (Operational Database) ยังไม่มีการเก็บข้อมูลย้อนหลัง (Historical Data) เพื่อใช้ช่วยในการคาดคะเนแนวโน้มที่คาดว่าจะเป็นไปได้ในอนาคต

2.2 คลังข้อมูล (Data Warehouse)

เป็นแหล่งเก็บรวบรวมข้อมูล เพื่อใช้ในการสนับสนุนการตัดสินใจของผู้บริหารองค์กรซึ่งเป็นหัวใจสำคัญ โดยมีลักษณะดังต่อไปนี้



ภาพประกอบ 2-1 สถาปัตยกรรมคลังข้อมูล (Architecture of data warehouses) [7]

2.2.1 การกำหนดทิศทางหรือมุ่งเน้นไปที่หัวข้อ (Subject-Oriented)

การกำหนดทิศทางหรือมุ่งเน้นไปที่หัวข้อ [2,3,4,5,6,7,8,11,14,17] เป็นการเก็บรวบรวมข้อมูลที่มีการมุ่งเน้นหรือกำหนดทิศทางไปที่หัวข้อหลัก ซึ่งเป็นข้อมูลขององค์กร มิใช่เป็นแอพลิเคชัน เช่น สินค้า การขาย ลูกค้า ผู้จำหน่าย เป็นต้น โดยมีจุดความสนใจอยู่ที่การสร้างและวิเคราะห์ข้อมูลเพื่อใช้ในการตัดสินใจของผู้บริหาร ไม่ใช่การประมวลผล ทรานแซคชันหรือเป็นการดำเนินงานเป็นรายวัน และจะกรองข้อมูลที่ไม่เกี่ยวข้องในการสนับสนุนการตัดสินใจออกไป ดังนั้นข้อมูลที่ถูกจัดเก็บและแสดงออกมาจึงอยู่ในรูปแบบที่ง่ายและกะทัดรัดตรงตามหัวข้อเรื่องที่ผู้ใช้สนใจดังภาพประกอบ 2-1

2.2.2 การรวมเข้าด้วยกันเป็นหนึ่งเดียว (Integrated)

การรวมเข้าด้วยกันเป็นหนึ่งเดียว [2,3,4,5,6,7,8,11,14,17] เนื่องจากคลังข้อมูลมีไว้สำหรับจัดเก็บรวบรวมข้อมูลจากหลาย ๆ แหล่งข้อมูลที่แตกต่างกัน เช่น ไฟล์ธรรมดา ฐานข้อมูลเชิงความสัมพันธ์ และรายการทรานแซคชันออนไลน์ (On-line Transaction Records) มารวมเข้าไว้ในแหล่งที่เดียวกัน ซึ่งข้อมูลแต่ละแหล่งก็ย่อมมีความแตกต่างกัน หรือไม่สอดคล้องกัน ในเรื่องของชื่อ โครงสร้างการเข้ารหัส หน่วยการวัดแอทริบิวต์ และอื่น ๆ เช่น หน่วยเงินตรา ค่าอาหารเช้า ภาษี เป็นต้น ดังนั้นเมื่อมีการย้ายข้อมูลไปยังคลังข้อมูล ก็ต้องมีเทคนิควิธีการล้างทำ

ความสะอาดข้อมูล และแปลงข้อมูล เพื่อให้เกิดความสอดคล้องตรงกันและเป็นไปในทิศทางเดียวกัน ดังภาพประกอบ 2-1

2.2.3 มีเวลาเข้ามาเกี่ยวข้อง (Time-Variant)

มีเวลาเข้ามาเกี่ยวข้อง [2,3,4,5,6,7,8,11,14,17] เวลาเป็นปัจจัยสำคัญของโครงสร้างหลักในคลังข้อมูล อาจจะเป็นแบบชัดแจ้งหรือไม่ชัดแจ้งก็ได้ ทั้งนี้เพราะคลังข้อมูลเป็นการจัดเก็บข้อมูลที่เป็นประวัติเอาไว้ โดยทั่วไปมักจะเป็นช่วงเวลา 5-10 ปีที่ผ่านมา ซึ่งจะตรงกันข้ามกับระบบฐานข้อมูลดำเนินการ (Operational Database System) ที่จะเป็นการเก็บรวบรวมข้อมูลที่เป็นปัจจุบัน ณ ขณะนั้น หรือมีการเก็บข้อมูลไว้ไม่เกิน 60 - 90 วัน ดังนั้นระบบฐานข้อมูลดำเนินการอาจจะมีเวลาเป็นองค์ประกอบด้วยหรือไม่ก็ได้ ดังภาพประกอบ 2-1

2.2.4 ไม่เปลี่ยนแปลงได้ง่ายหรือข้อมูลมีความเสถียร (Nonvolatile)

ไม่เปลี่ยนแปลงได้ง่ายหรือข้อมูลมีความเสถียร [2,3,4,5,6,7,8,11,14,17] สภาพแวดล้อมของคลังข้อมูลต้องการเพียงแค่การดำเนินการในการเข้าถึงข้อมูล 2 อย่างเท่านั้น คือ การเริ่มต้นโหลดข้อมูลและการเข้าถึงข้อมูล ดังนั้นจึงไม่จำเป็นต้องมีกลไกในการควบคุมการประมวลผลรายการข้อมูล การกู้คืน และการเกิดขึ้นพร้อมกัน เช่น การอ่านและการเขียน โดยปกติการเข้าถึงคลังข้อมูล มักจะเป็นการอ่านเพียงอย่างเดียว ส่วนการปรับปรุงข้อมูลให้ทันสมัยนั้นมักจะเป็นการเพิ่มเข้าไปในคลังข้อมูลจะไม่อยู่ในลักษณะที่เป็นการลบหรือแก้ไขข้อมูลในคลังข้อมูล ดังนั้นข้อมูลในคลังข้อมูลจึงเป็นข้อมูลที่เสถียรดังภาพประกอบ 2-1

การเปรียบเทียบความแตกต่างระหว่างระบบฐานข้อมูลดำเนินการและคลังข้อมูลแบ่งตามลักษณะต่าง ๆ สามารถพิจารณาได้ดังตาราง 2-1 [7]

ตาราง 2-1 แสดงการเปรียบเทียบฐานข้อมูลดำเนินการและคลังข้อมูล [7]

ลักษณะ	ฐานข้อมูลดำเนินการ	คลังข้อมูล
ข้อมูล	- ข้อมูลปัจจุบันที่ทันสมัย - ข้อมูลมีการเปลี่ยนแปลงได้ - มุ่งประเด็นนำข้อมูลดิบ(Data) เข้า	- ข้อมูลในอดีตจนถึงปัจจุบันที่มีความถูกต้องแม่นยำในช่วงเวลานาน ๆ - ข้อมูลไม่มีการเปลี่ยนแปลง - มุ่งประเด็นนำข้อมูล (Information)ออก
ผู้ใช้ระบบ	- เสมียน นักธุรกิจ ผู้ออกแบบระบบ	- ผู้จัดการ ผู้บริหาร นักวิเคราะห์
การทำงาน	- การดำเนินงานประจำวัน เป็นการทำงานแบบซ้ำ ๆ ใช้เวลาไม่นาน	- การดำเนินงานวิเคราะห์ข้อมูลเพื่อสนับสนุนการตัดสินใจ จึงต้องการข้อมูลในช่วงระยะเวลานาน
การมอง (View)	- แสดงรายละเอียดหน่วยเล็กที่สุด - โครงสร้างข้อมูลไม่เปลี่ยนแปลง แต่ข้อมูลที่อยู่ข้างในมีการเปลี่ยนแปลงได้	- แสดงข้อมูลที่มีการสรุปไว้ในลักษณะหลายมิติ - โครงสร้างมีการเปลี่ยนแปลงได้ แต่ข้อมูลที่อยู่ข้างในไม่มีการเปลี่ยนแปลง
การกำหนดทิศทาง	- มุ่งเน้นไปที่ลูกค้าเป็นสำคัญ	- มุ่งเน้นด้านการตลาด และหัวเรื่องในการวิเคราะห์เป็นสำคัญ
ขนาดของฐานข้อมูล	- 100 MB ถึง GB	- 100 GB ถึง TB
การออกแบบฐานข้อมูล	- ใช้แบบจำลองอี-อาร์ (E-R Model)	- ใช้แบบจำลองหลายมิติ (Multidimensional Model)
การประมวลผล	- แบบ OLTP	- แบบ OLAP
การเข้าถึง	- อ่าน/เขียน - ต้องมีกลยุทธ์ในการควบคุมความสอดคล้องตรงกัน และกู้คืน - จำนวนเรคอร์ดที่ถูกเข้าถึงเป็นสิบ - จำนวนผู้เข้าถึงข้อมูลเป็นพัน	- ส่วนมากเป็นการอ่าน - ไม่ต้องมีกลยุทธ์ในการควบคุมความสอดคล้องตรงกันและกู้คืน - จำนวนเรคอร์ดที่ถูกเข้าถึงเป็นล้าน - จำนวนผู้เข้าถึงข้อมูลเป็นร้อย

ตาราง 2-1 แสดงการเปรียบเทียบฐานข้อมูลดำเนินการและคลังข้อมูล (ต่อ)

ลักษณะ	ฐานข้อมูลดำเนินการ	คลังข้อมูล
มาตรวัดประสิทธิภาพ	- ปริมาณทรานแซคชัน	- ปริมาณการสอบถาม เน้นเวลาในการตอบสนอง
การสอบถาม	- เป็นการสอบถามแบบสั้น ๆ ง่าย ๆ - มีการจัดเตรียมไว้ล่วงหน้า	- เป็นการสอบถามที่ซับซ้อน - เป็นแบบทันทีทันใด ไม่ทราบล่วงหน้าว่าคำถามจะเป็นเช่นไร
การสรุปความ	- ง่าย ๆ ไม่ซับซ้อน ต้องการรายละเอียดมากและมีความถูกต้องแม่นยำในขณะที่มีการเข้าถึง	- มีการสรุปข้อมูลในอดีตไว้ให้ใช้และสามารถแสดงข้อมูลในอดีตได้อย่างรวดเร็ว

เราจะพบว่า คลังข้อมูลและฐานข้อมูลดำเนินการมีความแตกต่างกันหลายประการ เพิ่มความสามารถในการทำงานของทั้งสองระบบ เนื่องจากความแตกต่างทางด้านโครงสร้างและสิ่งที่บรรจุอยู่ข้างใน โดยฐานข้อมูลดำเนินการถูกออกแบบและปรับให้สอดคล้องกับภาระงาน ซึ่งเป็นเหตุผลหลักที่เราจะต้องแยกทั้งสองระบบออกจากกัน การจัดเตรียมความสามารถในการทำดัชนี และมีการค้นหาข้อมูลเป็นเรคอร์ด ซึ่งการสอบถามได้ถูกจัดเตรียมไว้ล่วงหน้าก่อนแล้ว ส่วนในคลังข้อมูลนั้น การสอบถามจะมีความซับซ้อนและเป็นแบบทันทีทันใด และต้องมีการเข้าถึงข้อมูลเป็นล้านเรคอร์ด นอกจากนั้นยังต้องมีการสแกนและการใช้เงื่อนไขที่ซับซ้อนเป็นจำนวนมาก ซึ่งในบางครั้งต้องมีวิธีการพัฒนาระบบ การเข้าถึง และการจัดการแบบพิเศษบนข้อมูลที่มีลักษณะเป็นหลายมิติ เพื่อให้การประมวลผลของระบบคลังข้อมูลมีประสิทธิภาพมากขึ้น

2.3 การค้นหาข้อมูล (Searching)

การค้นหาข้อมูลที่เราต้องการนั้น สามารถทำได้หลายวิธีด้วยกัน [1,12,15] เช่น

- การเข้าถึงแบบตามลำดับและใช้ลิสต์ หรือตาราง หรืออาร์เรย์
- การเข้าถึงโดยตรงโดยใช้ค่าคีย์ ซึ่งเรียกว่าการแฮช (Hashing)
- วิธีการทำดัชนี (Indexing Method)

แต่ในที่นี้จะขอกล่าวถึงเฉพาะวิธีที่ 3 เท่านั้น เพราะเลือกใช้วิธีการทำดัชนีในการค้นหาข้อมูลในวิทยานิพนธ์นี้

2.3.1 ดัชนี (Index)

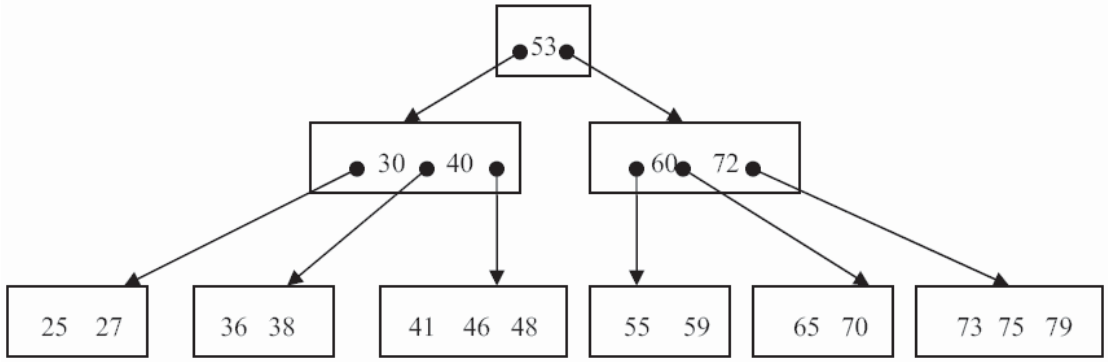
ดัชนี [1,2,3,4,9,11,12,14,16] เป็นวิธีการที่ช่วยทำให้เราสามารถค้นหาข้อมูลหรือหาแหล่งที่ตั้งของข้อมูลที่ตรงกับเงื่อนไขที่เราต้องการได้อย่างรวดเร็ว โดยดัชนีจะถูกเก็บบันทึกไว้ในอีกแหล่งหนึ่งที่ประกอบด้วยคีย์และตัวชี้ (key/pointer) ที่สามารถเข้าถึงเรคอร์ดข้อมูลได้โดยตรง รายการของดัชนีจะสั้นและเล็กกว่าเรคอร์ดข้อมูลจริงที่เก็บไว้มาก เป็นการลดการเข้าถึงอุปกรณ์ I/O ได้อย่างมาก เพราะถ้าไม่มีดัชนี ก็จะต้องสแกนตารางทั้งตาราง เพื่อหาแหล่งที่ตั้งของข้อมูลที่เราต้องการ ซึ่งเป็นวิธีที่ช้าและไม่มีประสิทธิภาพ ดัชนีที่มีประสิทธิภาพ จะสามารถทำการประมวลผลการสอบถามข้อมูลที่ซับซ้อนได้อย่างรวดเร็วกว่าดัชนีทั่วไป ซึ่งชนิดของดัชนีจะเป็นปัจจัยสำคัญกว่าจำนวนของดัชนีที่มีอยู่ในตาราง ต่อประสิทธิภาพของการประมวลผลการสอบถามข้อมูล

อย่างไรก็ตาม การสร้างดัชนีก็ย่อมต้องการใช้พื้นที่ในการจัดเก็บดัชนี ซึ่งเป็นเรื่องที่จะต้องพิจารณาในการจัดสรรทรัพยากรของฐานข้อมูลด้วย ดัชนีสามารถทำได้หลายวิธี ดังนี้

2.3.1.1 B-Tree Index

เป็นดัชนีที่จัดเก็บข้อมูลด้วยโครงสร้างแบบต้นไม้ [1,2,3] โดยอยู่ในรูปแบบของ Balance Tree ซึ่งประกอบด้วย โหนดบนสุด เรียกว่า Root Node โหนดต่อมา เรียกว่า Branch Node และ โหนดสุดท้าย เรียกว่า Leaf Node

สำหรับโครงสร้างทั่วไปของดัชนีแบบ B-Tree คือ โหนดภายในที่มี t คีย์ สามารถมีโหนดลูกได้ทั้งหมด $t+1$ โหนด ทุกคีย์ที่อยู่บนโหนดจะเรียงจากน้อยไปหามากและมีพอยน์เตอร์ระหว่างคีย์เป็นตัวจัดการแบ่งช่วงค่าของคีย์ได้ $t+1$ ช่วงค่า โหนดลูกที่เชื่อมมาจากพอยน์เตอร์ทางซ้ายของคีย์ จะต้องมีค่าน้อยกว่าคีย์นั้น ส่วนโหนดลูกที่เชื่อมมาจาก



ภาพประกอบ 2-2 ตัวอย่างดัชนีแบบ B-Tree

พอยน์เตอร์ทางขวาของคีย์จะต้องมีค่ามากกว่าคีย์นั้น [20] พิจารณาตัวอย่างดัชนีแบบ B-Tree ดังภาพประกอบ 2-2

ระยะทางระหว่างรูทและแต่ละลีฟของดัชนีแบบ B-Tree โหนดจะเหมือนกัน การสแกนต้นไม้โดยไม่คำนึงถึงค่าคีย์เพื่อเข้าถึงลีฟเพจต่าง ๆ ในระดับเดียวกันจะใช้เวลาเท่ากัน ซึ่งการเข้าถึงต้นไม้แบบกว้างจะเร็วกว่าต้นไม้แบบลึก เพราะการท่องแต่ละระดับของต้นไม้จะต้องเข้าถึงเพจหรือบล็อกอื่น ๆ เพิ่มเติมด้วย ดังนั้นเวลาที่ใช้ในการค้นหาข้อมูล จะเป็นค่าใช้จ่ายในการท่องต้นไม้จากรูทไปยังลีฟโหนด และค่าใช้จ่ายในการท่องไปยังหมายเลขเรคอร์ด

การจัดเก็บดัชนีในรูปแบบของ Balance Tree นี้จะช่วยให้การค้นหาข้อมูลมีความเร็วมากยิ่งขึ้น B-Tree เหมาะกับแอทริบิวต์ที่จะนำมาทำดัชนีที่มีค่าคาร์ดินอลิตี้สูง หรือข้อมูลที่มากกว่า 64 bit B-Trees ไม่เหมาะสำหรับการสอบถามข้อมูลที่ซับซ้อนและแอทริบิวต์ที่มีคาร์ดินอลิตี้ต่ำเพราะจะเสียค่าใช้จ่ายจะสูงมาก

2.3.1.1.1 ข้อดีของดัชนีแบบ B-Tree

ดัชนีแบบ B-Tree เป็นโครงสร้างข้อมูลที่ถูกออกแบบมาสำหรับทำงานเข้าถึงโดยตรงบนหน่วยความจำสำรอง มีการจัดการเกี่ยวกับ I/O ได้ดี ซึ่งระบบฐานข้อมูลส่วนใหญ่ก็มีการนำดัชนีแบบ B-Tree รวมถึงโครงสร้างอื่นที่อยู่บนพื้นฐานของดัชนีแบบ B-Tree เช่น B+ Tree และ B* Tree ไปใช้กับแอทริบิวต์ของข้อมูลที่มีลักษณะเป็นช่วงค่าหรือมีคาร์ดินอลิตี้สูง ๆ

2.3.1.1.2 ข้อจำกัดของดัชนีแบบ B-Tree

ดัชนีแบบ B-Tree ไม่เหมาะสำหรับการสอบถามที่ซับซ้อนและแอทริบิวต์มีคาร์ดินอลิตี้ต่ำ เนื่องจากจะต้องมีการสร้างคีย์ผสมขึ้นมาหากเป็นการสอบถามที่ต้องการข้อมูลมากกว่า 1 แอทริบิวต์ และเกิดความยุ่งยากและเสียค่าใช้จ่ายมากขึ้น

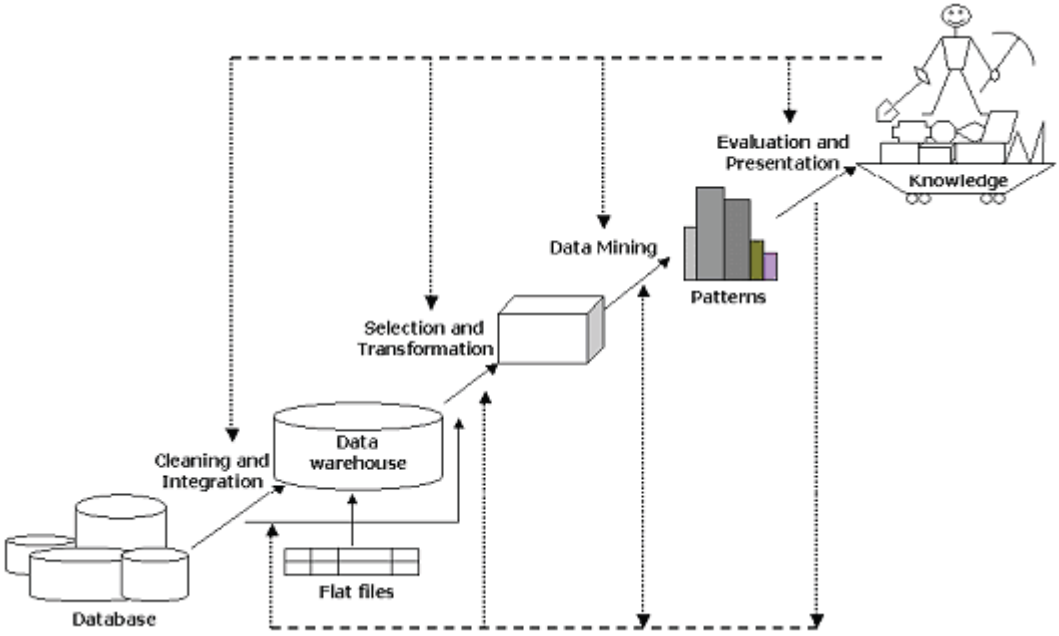
การสอบถามที่ซับซ้อนบนแอทริบิวต์ที่มีคาร์ดินอลิตี้ต่ำเกิดขึ้นบนคลังข้อมูลอยู่บ่อยครั้ง ซึ่งดัชนีที่เหมาะสำหรับการสอบถามในลักษณะนี้ คือดัชนีบิตแมป ซึ่งจะอธิบายในหัวข้อถัดไป

2.3.1.2 ดัชนีบิตแมป (Bitmap Index)

ดัชนีบิตแมป [1,2,3] เป็นดัชนีที่จัดเก็บข้อมูลในรูปแบบ Bitmap นั่นคือ เก็บข้อมูลในรูป 1 และ 0 เท่านั้นจึงมีความเร็วในการทำงานสูง ข้อมูลที่เหมาะกับการสร้างดัชนีแบบบิตแมปนั้นควรเป็นข้อมูลที่มีค่าคาร์ดินอลิตี้ต่ำ และมีการเปลี่ยนแปลงน้อย ดัชนีแบบบิตแมปสามารถกรองข้อมูลในระดับ WHERE clause ตัวอย่างของดัชนีบิตแมป เช่น แอทริบิวต์เพศ ก็จะมีค่าที่เป็นไปได้ 2 ค่า คือ เพศหญิง และเพศ ชาย ดังนั้นค่าคาร์ดินอลิตี้ของแอทริบิวต์เพศ ก็จะเท่ากับ 2 อย่างไรก็ตามดัชนีแบบบิตแมปจะไม่เหมาะกับข้อมูลที่มีจำนวนบิตเกิน 64 bit เพราะจะเสียค่าใช้จ่ายจะสูง

2.4 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล [5,6,7,10,11,13,17] เป็นการค้นหาความรู้ที่เป็นประโยชน์และน่าสนใจบนฐานข้อมูลขนาดใหญ่ (Knowledge Discovery from Very Large Databases: KDD) เมื่อใช้เทคนิคการทำเหมืองข้อมูล ข้อมูลขนาดใหญ่จะถูกวิเคราะห์และค้นหาความรู้หรือสิ่งที่สำคัญออกมาในรูปฐานความรู้ (Knowledge Base) เพื่อนำไปใช้ประโยชน์ต่อไป กระบวนการสืบค้นความรู้ ประกอบด้วยขั้นตอนการทำงานย่อยที่จะเปลี่ยนข้อมูลดิบให้กลายเป็นความรู้ใหม่ โดยเป็นกระบวนการสร้างแบบจำลองหรือกฎจากข้อมูลที่มีขึ้นมา เพื่อให้เกิดความเข้าใจในความสัมพันธ์ของข้อมูล และสามารถแยกประเภท ทำนายข้อมูลที่เป็นข้อมูลที่มีประโยชน์ต่อความต้องการของเราออกมาได้ ซึ่งประกอบด้วยลำดับขั้นตอนดังนี้ การกลั่นกรองข้อมูล (Data Cleaning) การรวบรวมข้อมูล (Data Integration) การคัดเลือกข้อมูล (Data Selection) การแปลงรูปข้อมูล (Data Transformation) การทำเดต้าไมนิ่ง (Data Mining) การประเมินรูปแบบหรือกฎที่ได้ (Pattern Evaluation) และ การนำเสนอความรู้ (Knowledge Presentation) โดยขั้นตอนข้างต้นสามารถวนกลับไปทำงานยังขั้นตอนที่ผ่านมาได้ สำหรับรายละเอียดของการทำงานอธิบายได้ดังนี้ (ดูภาพประกอบ 2-3)



ภาพประกอบ 2-3 กระบวนการการค้นหาความรู้

- การกลั่นกรองข้อมูล (Data Cleaning)

เป็นกำจัดข้อมูลขยะ ข้อมูลผิดปกติ ข้อมูลที่ไม่สมบูรณ์ เพื่อให้มั่นใจได้ว่าข้อมูลที่ได้นั้นถูกต้องและเหมาะสมต่อการทำเหมืองข้อมูล เพราะว่าข้อมูลทั้งหมดที่ได้มา อาจจะยังมีข้อผิดพลาดอยู่

- การรวบรวมข้อมูล (Data Integration)

เป็นการรวบรวมข้อมูลที่เป็นรูปแบบเดียวกันหรือใกล้เคียงกัน จากหลาย ๆแหล่งมารวมกันเป็นข้อมูลชุดเดียวกัน

- การคัดเลือกข้อมูล (Data Selection)

เป็นการระบุลักษณะข้อมูลที่ต้องการ แล้วคัดเลือกเอาข้อมูลที่ต้องการที่เกี่ยวข้องกับการวิเคราะห์ออกมา เฉพาะเรื่องที่สนใจตามจุดประสงค์ที่ตั้งเอาไว้

- การแปลงรูปข้อมูล (Data Transformation)

เป็นการแปลงข้อมูลให้อยู่ในรูปที่พร้อมจะนำไปวิเคราะห์ตามหลักของ data mining เช่น ข้อมูลอายุ จะเป็นข้อมูลเป็นตัว ๆ เช่น 17 36 28 อาจจะจัดเป็นกลุ่มช่วงอายุเพื่อสะดวกในการใช้งานและเกิดความเข้าใจมากขึ้น หรือ เทคนิคการแปลงกลุ่มประเภทให้เป็นตัวเลขเพื่อความสะดวก เช่น การใช้รหัสแทนชื่อของสิ่งของใด ๆ

ใน 4 ขั้นตอนข้างต้นนี้ อาจเรียกรวมกันได้เป็น การเตรียมข้อมูล (Data Preparation) ซึ่งเป็นขั้นตอนที่สำคัญมาก และใช้เวลานานที่สุด เพราะว่าหากเกิดข้อผิดพลาด อาจทำให้ผลที่ได้ออกมาคลาดเคลื่อน หรือผิดจากจุดประสงค์ที่วางไว้ไปมาก ดังนั้น อาจถือว่า การเตรียมข้อมูลเป็นหัวใจสำคัญของงาน

- การทำเดต้าไมนิ่ง (Data Mining)

เป็นการประมวลผลความสัมพันธ์ ตามอัลกอริทึมส์ที่ได้วางแผนเอาไว้ โดยจะนำเอาข้อมูลที่ผ่านขั้นตอนการเตรียมข้อมูลมาแล้ว มาใช้ในการประมวลผล โดยเมื่อทำขั้นตอนนี้แล้ว อาจมีการย้อนกลับไปทำขั้นตอนการเตรียมข้อมูลใหม่อีกครั้ง ซึ่งในขั้นตอนนี้ก็มีรูปแบบของอัลกอริทึมส์หลายแบบ

- การประเมินรูปแบบหรือกฎที่ได้ (Pattern Evaluation)

เป็นขั้นตอนการวิเคราะห์และประเมินผลข้อมูลที่ได้จากการทำไมนิ่ง ว่าได้ตรงตามความต้องการที่ตั้งเอาไว้หรือไม่ ข้อมูลที่ได้จะน่าสนใจแค่ไหน เพื่อนำไปใช้หรือแก้ไขต่อไป

- การนำเสนอความรู้ (Knowledge Presentation)

การนำข้อมูลที่ผ่านกระบวนการขั้นต้นแล้ว ไปนำเสนอในรูปแบบต่าง ๆที่เข้าใจง่าย และ นำเอาไปประยุกต์ใช้งานจริง หากว่าได้ผลลัพธ์เป็นที่น่าพึงพอใจตามจุดประสงค์ที่ตั้งไว้

อย่างไรก็ตาม การทำเหมืองข้อมูลเป็นเพียงเครื่องมือที่ใช้ในการวิเคราะห์เท่านั้น ไม่สามารถเข้าใจธุรกิจ หรือเข้าใจข้อมูลได้ดีเท่าคน โดยในปัจจุบันได้มีการนำเทคนิคการทำเหมืองข้อมูลไปประยุกต์ใช้ในงานด้านต่าง ๆ ซึ่งมีเทคนิคต่าง ๆดังนี้

- **Association Rule** คือ การค้นหาความสัมพันธ์ของข้อมูลจากข้อมูลขนาดใหญ่ที่มีอยู่เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่าง ๆ หรือมาจากการวิเคราะห์การซื้อสินค้าของลูกค้าเรียกว่า “Market Basket Analysis” ซึ่งประเมินจากข้อมูลในตารางที่รวบรวมไว้ ผลการวิเคราะห์ที่ได้จะเป็นคำตอบของปัญหา ซึ่งการวิเคราะห์แบบนี้เป็นการใช้ “กฎความสัมพันธ์” (Association Rule) เพื่อหาความสัมพันธ์ของข้อมูล
- **Classification** เป็นกระบวนการสร้าง model จัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ ตัวอย่างเช่น แบ่งประเภทของลูกค้าว่าเชื่อถือได้ หรือไม่โดยพิจารณาจากข้อมูลที่มีอยู่หรือ จัดกลุ่มนักเรียนว่า ดีมาก ดี ปานกลาง ไม่ดี โดยพิจารณาจากประวัติและผลการรียนเป็นต้น
- **Clustering** เป็นเทคนิคการลดขนาดของข้อมูลด้วยการแบ่งกลุ่มข้อมูล ที่มีลักษณะเดียวกันไว้ด้วยกัน และกลุ่มข้อมูลที่ไม่เหมือนกันแยกออกจากกัน

- **Deviation Detection** เป็นกรรมวิธีในการหาค่าที่แตกต่างไปจากค่ามาตรฐาน หรือค่าที่คาดคิดไว้ว่าต่างไปมากน้อยเพียงใด โดยทั่วไปมักใช้วิธีการทางสถิติ หรือการแสดงให้เห็นภาพ (Visualization) สำหรับเทคนิคนี้ใช้ในการตรวจสอบ ลายเซ็นปลอม หรือบัตรเครดิตปลอม รวมทั้งการตรวจหาจุดบกพร่องของชิ้นงานในโรงงานอุตสาหกรรม

งานวิจัยนี้จะขอกล่าวถึงเฉพาะวิธี Clustering เท่านั้น เพราะได้เลือกใช้วิธีดังกล่าวในวิทยานิพนธ์นี้

2.5 การแบ่งกลุ่มข้อมูล (Clustering)

การแบ่งกลุ่มข้อมูล (Data Clustering) [5,6,7,9,10,17] เป็นขั้นตอนเบื้องต้นของการวิเคราะห์ข้อมูล วิธีของการ Clustering นี้เป็นวิธีที่อาจจะเรียกว่าเป็นการทำ Data Mining แบบอ้อม ๆ ก็ว่าได้ เนื่องจากการหาผลลัพธ์ในแต่ละครั้งนั้น แม้กระทั่งผู้หายังไม่อาจจะทราบว่าสิ่งที่ต้องการจะหานั้นคืออะไร จำเป็นต้องรอจนกว่าการค้นหาจะทำเสร็จสมบูรณ์จึงจะทราบข้อมูลที่ซ่อนอยู่ เปรียบเสมือนกับการมีข้อมูลจำนวนมากมายอยู่ในตะกร้า แล้วจากนั้นก็มีบางสิ่งมาจัดเรียงข้อมูลหน่วยนั้นให้อยู่เป็นกลุ่มก้อน ซึ่งทำให้สังเกตลักษณะเด่นที่ซ่อนเร้นอยู่ภายในข้อมูลจำนวนมากนั้น

การแบ่งกลุ่ม (Clustering) จะแบ่งชุดข้อมูลออกเป็นกลุ่ม (Cluster) นำข้อมูลที่มีคุณลักษณะเหมือนกัน หรือคล้ายกันจัดไว้ในกลุ่มเดียวกัน ขั้นตอนวิธีที่ใช้ในการแบ่งกลุ่มจะอาศัยความเหมือน (Similarity) หรือความใกล้ชิด (Proximity) โดยคำนวณจากการวัดระยะระหว่างเวกเตอร์ของข้อมูล โดยใช้การวัดระยะแบบต่าง ๆ เช่น ฟังก์ชันระยะห่างยูคลีเดียน แมนฮัตตัน เป็นต้น

ในการแบ่งข้อมูลนั้นจะไม่มีการนิยามหมวดหมู่ (Classes) ก่อน และจะไม่มีตัวอย่างหมวดหมู่ แต่จะเป็นการเก็บบันทึกกลุ่มเดียวกัน ซึ่งดูจากพื้นฐานของความคล้ายคลึงกันของข้อมูลเอง และความคล้ายนั้นจะถูกกำหนดตามผู้ใช้งานที่จะเลือกให้ส่งผลเรื่องใด เพื่อยึดถือว่าเป็นผลของการแบ่งกลุ่ม

ตัวอย่างเช่น การแบ่งกลุ่มอาจเป็นขั้นตอนแรกในความพยายามของการแบ่งในส่วนการตลาด แทนที่จะพยายามแนะนำ one-size-fits เหมาะกับทุกกฎสำหรับ “promotion แบบใดที่จะได้รับการตอบสนองจากลูกค้าดีที่สุด” เริ่มแรกแบ่งพื้นฐานลูกค้าเข้าไปในกลุ่ม หรือประชากรด้วยนิสัยที่คล้ายกันในการซื้อ แล้วก็ลองถามดูว่า promotion แบบใดที่จะเหมาะสมและดีที่สุดสำหรับแต่ละกลุ่ม

ขั้นตอนวิธีการพิจารณาการแบ่งกลุ่มข้อมูล สามารถแยกออกได้หลายประเภท ยกตัวอย่างเช่น กรณีที่ทราบจำนวนกลุ่มที่ต้องการแบ่งกลุ่มข้อมูล หรือ กรณีที่ไม่ทราบจำนวนกลุ่มที่ต้องการแบ่งกลุ่มข้อมูล เป็นต้น ตัวอย่างการแบ่งกลุ่มข้อมูล เช่น Partitioning Clustering Hierarchical Clustering Density-based Clustering และ Grid-based Clustering เป็นต้น

งานวิจัยนี้จะทราบจำนวนกลุ่มที่ต้องการแบ่งกลุ่มข้อมูลชัดเจน ดังนั้นจะขอกล่าวถึงเฉพาะวิธี Partitioning Clustering [5,7,10,17] ซึ่งวิธีนี้จะแบ่งกลุ่มข้อมูลให้มีจำนวนเท่ากับจำนวนกลุ่มที่ต้องการ โดยที่ทุกกลุ่มข้อมูลไม่มีการซ้อนทับกัน Partitioning Clustering จะมีขั้นตอนการทำงานพื้นฐานดังนี้

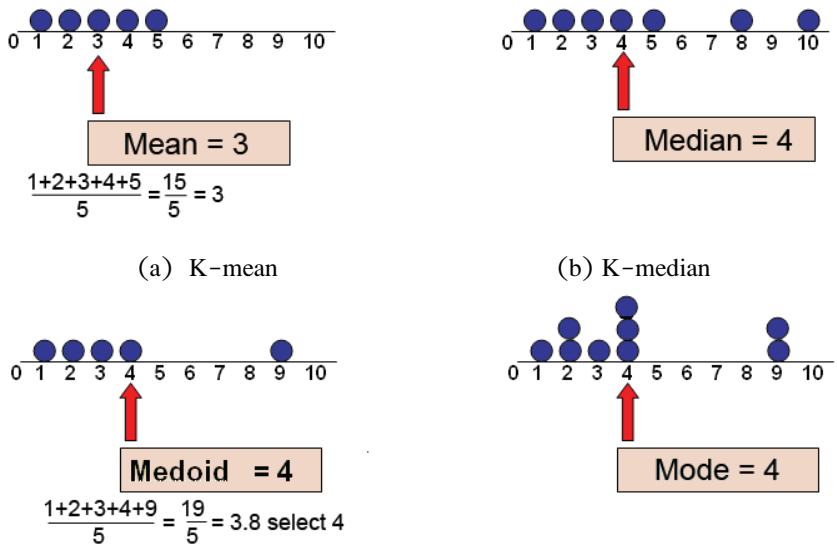
1.ทำการหาข้อมูล K ตัวซึ่งใช้เป็นค่าศูนย์กลางหรือตัวแทนกลุ่มในแต่ละกลุ่มของ K กลุ่ม

2.ทำการอ่านข้อมูลจากฐานข้อมูล จัดให้ข้อมูลแต่ละตัวอยู่ในกลุ่มที่มีค่าระยะห่างกันน้อยที่สุด (เหมือนกันมากที่สุด)

3. หาค่าศูนย์กลางกลุ่มของแต่ละกลุ่มใหม่ จำนวน K กลุ่ม

4. ทำการหากลุ่มให้ข้อมูลทุกตัวจากศูนย์กลางใหม่ที่ได้

5.ทำซ้ำในขั้นตอนที่ 3 และ 4 จนกว่าข้อมูลทุกตัวจะไม่มีการเปลี่ยนกลุ่ม



ภาพประกอบ 2-4 รูปแบบต่าง ๆ ในการเลือกค่าศูนย์กลางหรือตัวแทนกลุ่ม

Partitioning Clustering มีอัลกอริทึมมากมายซึ่งแต่ละอัลกอริทึม จะมีขั้นตอนการทำงานพื้นฐานเดียวกันแต่จะแตกต่างตรงการเลือกค่าศูนย์กลาง หรือตัวแทนกลุ่ม (ขั้นตอนการทำงานที่ 1 และ 3 ของขั้นตอนการทำงานพื้นฐาน) ซึ่งมีรูปแบบต่าง ๆ ดังนี้ (ดูภาพประกอบ 2-4)

2.5.1 K-Mean Clustering

เป็นการนำค่าเฉลี่ยของข้อมูล [5, 7, 9, 10, 17] มาเป็นค่าศูนย์กลางหรือตัวแทนกลุ่ม ตัวอย่างเช่น ดังภาพประกอบ 2-4 (a) นำข้อมูลทั้ง 5 ค่ามาหาค่าเฉลี่ยจะได้ว่า ข้อมูลดังกล่าวมีค่าเฉลี่ยเท่ากับ 3 นำค่า 3 เป็นศูนย์กลางหรือตัวแทนกลุ่ม

2.5.2 K-Median Clustering

เป็นการนำค่ากึ่งกลางของข้อมูล [5, 7, 17] ที่ถูกจัดเรียงลำดับแล้วมาเป็นค่าศูนย์กลางหรือตัวแทนกลุ่ม ตัวอย่างเช่น ดังภาพประกอบ 2-4(b) นำข้อมูลทั้ง 7 มาจัดเรียงข้อมูลตามลำดับ จากนั้นแยกข้อมูลออกให้เท่า ๆกันและนำค่าตรงกลางมาเป็นค่าศูนย์กลางหรือตัวแทนกลุ่ม

2.5.3 K-Medoid Clustering

เป็นการนำค่าจริงของข้อมูล [5, 7, 17] ที่มีค่าใกล้ค่าเฉลี่ยมาเป็นค่าศูนย์กลางหรือตัวแทนกลุ่ม ตัวอย่างเช่น ดังภาพประกอบ 2-4 (c) นำข้อมูลทั้ง 5 ค่ามาหาค่าเฉลี่ยจะได้ว่า ข้อมูลดังกล่าวมีค่าเฉลี่ยเท่ากับ 3.8 แต่ค่า 3.8 ไม่มีค่าจริง ดังนั้น จะนำค่าจริงที่มีอยู่นั้นคือค่า 4 เป็นศูนย์กลางหรือตัวแทนกลุ่ม

2.5.4 K-Mode Clustering

ลักษณะของข้อมูลที่ใช้ใน K-Mode จะเป็นข้อมูลที่นำมาคำนวณไม่ได้ หรือที่วัดค่าได้ยาก เช่น สี เพศ เป็นต้น ดังนั้น K-Mode จะนำค่าที่มีความถี่มากที่สุดมาเป็นค่าศูนย์กลางหรือตัวแทนกลุ่ม [5,7,10,17] ตัวอย่างเช่น ดังภาพประกอบ 2-4 (d) ค่าที่มีความถี่มากที่สุดคือ ค่า 4 ดังนั้น จะนำค่า 4 เป็นศูนย์กลางหรือตัวแทนกลุ่ม

งานวิจัยชิ้นนี้ ได้เลือกใช้วิธีการ K-mode Clustering มาช่วยในการแบ่งกลุ่มข้อมูลของดัชนีบิตแมปแบบกระจาย เพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิก เนื่องจากวิธีดังกล่าว เหมาะกับดัชนีบิตแมปแบบกระจายซึ่งข้อมูลในแอทริบิวต์ที่นำมาสร้างดัชนีดังกล่าวนั้น มักเป็นข้อมูลที่ไม่เปลี่ยนแปลง ข้อมูลที่ไม่สามารถนำมาคำนวณได้ และ งานวิจัยชิ้นนี้ทราบจำนวนกลุ่มที่ต้องการแบ่งกลุ่มอย่างชัดเจนก่อนทำการแบ่งกลุ่มข้อมูล

บทที่ 3

ดัชนีแบบบิตแมป (Bitmap Index)

จากบทที่ 2 ที่ได้กล่าวถึงระบบฐานข้อมูล และ อธิบายเกี่ยวกับการเข้าถึงข้อมูลโดยใช้ดัชนีไว้ข้างต้น ดัชนีที่นิยมใช้บนระบบฐานข้อมูลดำเนินการ คือ ดัชนีแบบ B-Tree ซึ่งเหมาะกับแอทริบิวต์ที่จะนำมาทำดัชนีที่มีค่าคาร์ดินอลิตี้สูง (จำนวนค่าที่เป็นไปได้ของแอทริบิวต์) ไม่เหมาะสำหรับการสอบถามข้อมูลที่ซับซ้อนและแอทริบิวต์ที่มีคาร์ดินอลิตี้ต่ำเพราะจะเสียค่าใช้จ่ายจะสูงมาก สำหรับบนคลังข้อมูลก็มีใช้เช่นกัน แต่ในกรณีที่แอทริบิวต์มีคาร์ดินอลิตี้ต่ำมักเลือกใช้ดัชนีบิตแมปแทน เช่น แอทริบิวต์เพศ แอทริบิวต์จังหวัด เป็นต้น

สำหรับบทนี้ จะเป็นการกล่าวถึงการทำดัชนีที่อยู่บนพื้นฐานของบิตแมป ซึ่งได้แก่ ดัชนีบิตแมปแบบพื้นฐาน ดัชนีบิตแมปแบบช่วง ดัชนีบิตแมปแบบเข้ารหัส และดัชนีบิตแมปแบบกระจาย

ดัชนีบิตแมปเป็นดัชนีที่จัดเก็บข้อมูลในรูปแบบ Bitmap นั่นคือ เก็บข้อมูลในรูป 1 และ 0 เท่านั้น เนื่องจากมีการดำเนินการระดับบิต (AND OR XOR NOT) ระหว่างบิตแมปเวกเตอร์ก่อนดึงข้อมูลจริง ทำให้มีประสิทธิในการประมวลผลเพราะสนับสนุนการทำงานของฮาร์ดแวร์ จึงมีความเร็วในการทำงานสูง ข้อมูลที่เหมาะกับการสร้างดัชนีแบบบิตแมปนั้นควรเป็นข้อมูลที่มีค่าคาร์ดินอลิตี้ต่ำๆ และมีการเปลี่ยนแปลงน้อย ดัชนีแบบบิตแมปสามารถกรองข้อมูลในระดับ WHERE clause ตัวอย่างของดัชนีบิตแมป เช่น แอทริบิวต์เพศ ก็จะมีค่าที่เป็นไปได้ 2 ค่า คือ เพศหญิง และเพศชาย ดังนั้นค่าคาร์ดินอลิตี้ของแอทริบิวต์เพศ ก็จะเท่ากับ 2 อย่างไรก็ตามดัชนีแบบบิตแมปจะไม่เหมาะกับข้อมูลที่มีจำนวนบิตเกิน 64 bit เพราะจะเสียค่าใช้จ่ายจะสูง

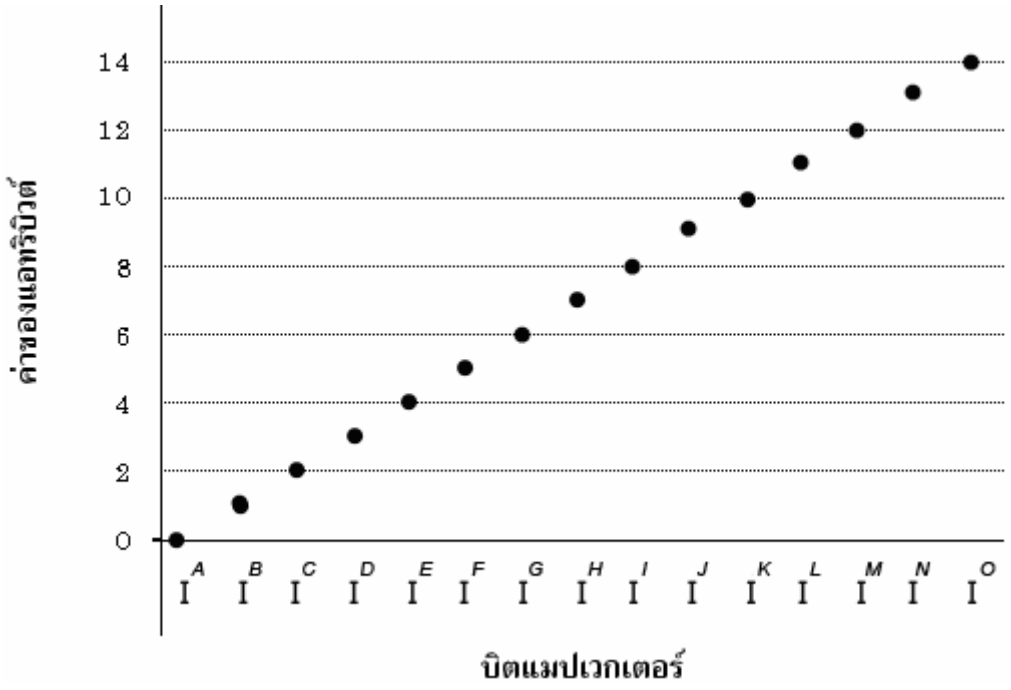
งานวิจัยที่ผ่านมามีนำเสนอเทคนิคการทำดัชนีบิตแมปที่น่าสนใจไว้หลายวิธี ซึ่งในบทนี้เราจะกล่าวถึง 4 ชนิด ได้แก่

- ดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index) [2]
- ดัชนีบิตแมปแบบช่วง (Interval Bitmap Index) [2]
- ดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index) [4]
- ดัชนีบิตแมปแบบกระจาย (Scatter Bitmap Index) [3]

3.1 ดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index)

ดัชนีบิตแมปแบบพื้นฐาน [2] เป็นดัชนีบิตแมปที่มีแอทริบิวต์ที่มีคาร์ดินอลิตี้เท่ากับ C ประกอบด้วย C บิตแมปเวกเตอร์ เช่น บิตแมปเวกเตอร์ $I^A, I^B, ..., I^O$ โดย I^i คือ บิตแมปเวกเตอร์ที่แทนค่าของแอทริบิวต์ที่มีค่าเท่ากับ i เช่น I^D เป็นบิตแมปเวกเตอร์ที่แทนค่าของแอทริบิวต์ที่มีค่าเท่ากับ D กล่าวคือ ดัชนีบิตแมปแบบพื้นฐานมีการแทนค่าข้อมูลโดยใช้ 1 บิตแมปเวกเตอร์ต่อ 1 ค่าของแอทริบิวต์

จากภาพประกอบ 3-1 แสดงการทำดัชนีบิตแมปแบบพื้นฐานบนแอทริบิวต์ Y ซึ่งมีค่าที่เป็นไปได้ 15 ค่า คือ A B C D E F G H I J K L M N และ O ดังนั้น C = 15 จึงใช้ 15 บิตแมปเวกเตอร์ นั่นคือ $I^A\ I^B\ ..I^O$



ภาพประกอบ 3-1 แผนภาพการลงรหัสดัชนีบิตแมปแบบพื้นฐานบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 15 บิตแมปเวกเตอร์

RID	แอทริบิวต์ Y	I^O	I^N	...	I^F	I^E	I^D	I^C	I^B	I^A
1	H	0	0	...	0	0	0	0	0	0
2	D	0	0	...	0	0	1	0	0	0
3	L	0	0	...	0	0	0	0	0	0
4	C	0	0	...	0	0	0	1	0	0
5	D	0	0	...	0	0	1	0	0	0
6	B	0	0	...	0	0	0	0	1	0
7	N	0	1	...	0	0	0	0	0	0
8	A	0	0	...	0	0	0	0	0	1
9	G	0	0	...	0	0	0	0	0	0
:	:	:	:	...	:	:	:	:	:	:
100	F	0	0	...	1	0	0	0	0	0

ภาพประกอบ 3-2 ดัชนีบิตแมปแบบพื้นฐานบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 15 บิตแมปเวกเตอร์

จากภาพประกอบ 3-2 เรคอร์ดที่ 2 และ 5 ของแอทริบิวต์ Y มีค่าเท่ากับ D ดังนั้น บิตที่ 1 บิตแมปเวกเตอร์ I^D จึงมีค่าเท่ากับ 1 ส่วนบิตอื่นที่เหลือของบิตแมปเวกเตอร์ I^v มีค่าเท่ากับ 0

3.1.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบพื้นฐาน

สำหรับการค้นหาข้อมูลแบบค่าเท่ากัน (Equality query) มีรูปแบบการสอบถามแบบค่าเท่ากัน คือ “Y = v” หมายถึง การสอบถามว่า บนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ v สำหรับการสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบพื้นฐาน สามารถค้นหาโดยการอ่านค่าบิตแมปเวกเตอร์นั้น ๆ (I^v)

ตัวอย่างการสอบถาม พิจารณาจากภาพประกอบ 3-2 ต้องการค้นหาค่าของ Y= D หาได้โดยตรวจสอบบิตแมปเวกเตอร์ I^D บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จะได้ว่า แถวที่ 2 และ 5 เป็นคำตอบของการสอบถาม

3.1.2 การสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบพื้นฐาน

มีรูปแบบ “Y in { v_1, v_2,.., v_k }” หมายถึงการสอบถามว่า บนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ v สำหรับการสอบถามแบบค่าสมาชิกบนดัชนีบิตแมปแบบพื้นฐานสามารถทำได้โดยการอ่านค่าบิตแมปเวกเตอร์ที่สัมพันธ์กับค่า v ของค่าที่ต้องการหาทุก ๆค่า (I^v) แล้วนำผลที่ได้แต่ละค่ามาดำเนินการตรรกะ OR ($\vee$)กันระหว่างบิตแมปเวกเตอร์

ตัวอย่างการสอบถามพิจารณา จากภาพประกอบ 3-2 ต้องการค้นหาค่าของ Y in (D, F, H, L, N) สามารถทำได้ดังนี้ $I^D \vee I^F \vee I^H \vee I^L \vee I^N$ จะได้ว่าแถวที่ 1 2 3 5 7 และ 100 เป็นคำตอบของการสอบถาม

3.1.3 ข้อดีของดัชนีบิตแมปแบบพื้นฐาน

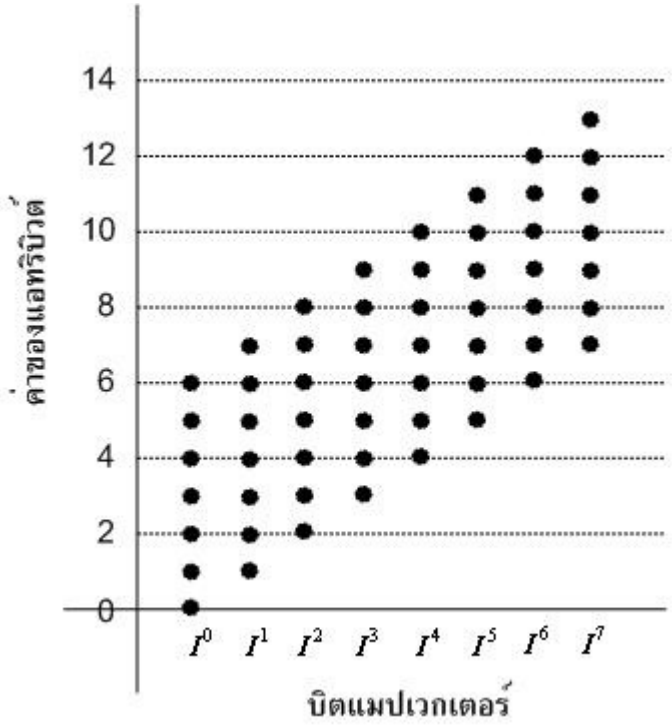
ดัชนีบิตแมปแบบพื้นฐานจะเหมาะสำหรับการสอบถามข้อมูลแบบค่าเท่ากัน เนื่องจากจะมีการอ่านบิตแมปเวกเตอร์เพียง 1 บิตแมปเวกเตอร์ และไม่มีการดำเนินการตรรกะใด ๆ เกิดขึ้น กล่าวคือ การสอบถามข้อมูลแบบค่าเท่ากันจะใช้เวลาน้อยในการค้นหาข้อมูล

3.1.4 ข้อจำกัดของดัชนีบิตแมปแบบพื้นฐาน

อย่างไรก็ตามดัชนีบิตแมปแบบพื้นฐาน จะไม่เหมาะสำหรับแอทริบิวต์ที่มีค่าคาร์ดินอลิตี้สูง ๆ เมื่อคาร์ดินอลิตี้ของแอทริบิวต์เพิ่มสูงขึ้น จะทำให้พื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปแบบพื้นฐานมากขึ้นและใช้ประโยชน์ของพื้นที่ได้ไม่เต็มที่ ซึ่งปัญหาที่แอทริบิวต์มีคาร์ดินอลิตี้สูงขึ้นนี้เอง ได้นำไปสู่การคิดค้นเทคนิคการทำดัชนีบิตแมปแบบอื่น ๆ ตามมา

3.2 ดัชนีบิตแมปแบบช่วง (Interval Bitmap Index)

จากหัวข้อ 3.1 ที่กล่าวมาข้างต้น จะพบข้อจำกัดของการทำดัชนีบิตแมปแบบพื้นฐานเมื่อมีแอทริบิวต์ที่มีค่าคาร์ดินอลิตี้สูง ๆ จึงมีนักวิจัยได้คิดค้นและนำไปสู่การคิดค้นดัชนีบิตแมปแบบช่วง [2] ที่ใช้พื้นที่ในการจัดเก็บดัชนีน้อยกว่าเดิม กล่าวคือ ดัชนีบิตแมปแบบช่วงจะใช้พื้นที่ในการจัดเก็บดัชนี $\lceil C/2 \rceil$ บิตแมปเวกเตอร์ คือ $I^0, I^1, ..., I^{\lceil C/2 \rceil - 1}$ โดยที่ I^j แทนบิตแมปเวกเตอร์ที่มีค่าของแอทริบิวต์อยู่ในช่วง $I^j = [j, j+m]$ และ $m = \lfloor C/2 \rfloor - 1$ ซึ่งในการแทนค่าจะกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ I^j มีค่าเป็น 1 ก็ต่อเมื่อเรคอร์ดที่ i มีค่าของแอทริบิวต์อยู่ในช่วง $[j, j+m]$ ส่วนบิตอื่นที่เหลือมีค่าเป็น 0 พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบช่วงดังภาพประกอบ 3-3



ภาพประกอบ 3-3 แผนภาพการลงรหัสดัชนีบิตแมปแบบช่วงบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-3 แอทริบิวต์ Y มีค่าที่เป็นไปได้ 15 ค่า คือ 0,1,2,...,14 ดังนั้นดัชนีบิตแมปแบบช่วงประกอบด้วย 8 $\left(=\lceil 15/2 \rceil\right)$ บิตแมปเวกเตอร์ คือ $I^0, I^1, ..., I^7$ และ $m = \lfloor 15/2 \rfloor - 1 = 6$ ดังนั้น บิตแมปเวกเตอร์ I^j แทนค่าของแอทริบิวต์ Y ที่อยู่ในช่วง $[j, j+6]$ หรือ $I^j = [j, j+6]$ จึงได้ว่า

$$I^0 = [0,6], \quad I^1 = [1,7], \quad I^2 = [2,8], \quad I^3 = [3,9],$$
$$I^4 = [4,10], \quad I^5 = [5,11], \quad I^6 = [6,12], \quad I^7 = [7,13]$$

พิจารณาตัวอย่างดัชนีบิตแมปแบบช่วงดังภาพประกอบ 3-4

RID	แอทริบิวต์ Y	I^7	I^6	I^5	I^4	I^3	I^2	I^1	I^0
1	3	0	0	0	0	1	1	1	1
2	2	0	0	0	0	0	1	1	1
3	3	0	0	0	0	1	1	1	1
4	1	0	0	0	0	0	0	1	1
5	13	1	0	0	0	0	0	0	0
6	4	0	0	0	1	1	1	1	1
7	6	0	1	1	1	1	1	1	1
8	5	0	0	1	1	1	1	1	1
9	14	0	0	0	0	0	0	0	0
:	:	:	...	:	:	:	:	:	:
100	2	0	0	0	0	0	1	1	1

ภาพประกอบ 3-4 ดัชนีบิตแมปแบบช่วงบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-4 เรคอร์ดที่ 1 ของแอทริบิวต์ Y มีค่าเท่ากับ 3 ซึ่งอยู่ในช่วงของบิตแมปเวกเตอร์ I^0, I^1, I^2 และ I^3 ดังนั้น บิตที่ 1 ของบิตแมปเวกเตอร์ I^0, I^1, I^2 และ I^3 จึงมีค่าเท่ากับ 1 ส่วนบิตอื่นที่เหลือของบิตแมปเวกเตอร์ v มีค่าเท่ากับ 0

3.2.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบช่วง

การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบช่วงมีรูปแบบทั่วไปคือ

$$\text{“Y = v =”} \begin{cases} I^0 & \text{if} \quad v=0,\ m=0, \\ \overline{I^0} & \text{if} \quad v=1, C=2, \\ I^1 & \text{if} \quad v=1, C=3, \\ I^v \wedge \overline{I^{v+1}} & \text{if} \quad v<m, \\ I^v \wedge I^0 & \text{if} \quad v=m, m>0, \\ I^{v-m} \wedge \overline{I^{v-m-1}} & \text{if} \quad \mathrm{m}<v<C-1, m>0, \\ \overline{(I^{\lceil \frac{C}{2} \rceil -1} \vee I^0)} & \text{if} \quad v=C-1 \end{cases}$$

ตัวอย่างเช่น จากภาพประกอบ 3-4 ต้องการทราบว่า บนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 3 ซึ่งตรงกับเงื่อนไขที่ 4 ($v < m$ ซึ่งก็คือ $3 < 6$) จะได้ว่าคำตอบสามารถหาได้จาก $"Y = 3" = I^3 \wedge \overline{I^4}$ ซึ่งแบ่งเป็นขั้นตอนเพื่อหาคำตอบได้ดังนี้

1) อ่านบิตแมปเวกเตอร์ I^3

2) อ่านบิตแมปเวกเตอร์ I^4

3) ดำเนินการตรรกะ NOT ทุกบิตบนบิตแมปเวกเตอร์ I^4 (ได้จากข้อ 2)

4) ดำเนินการตรรกะ AND บิตต่อบิตระหว่างบิตแมปเวกเตอร์ผลลัพธ์ที่ได้จากข้อ 1) และ 3)

5) คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้จากข้อ 4 พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น 3 คือเรคอร์ดที่ 1 และ 3

3.2.2 การสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบช่วง

การสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบช่วงสามารถค้นหาโดยการอ่านค่าบิตแมปเวกเตอร์ ตามเงื่อนไขการค้นหาข้อมูลแบบค่าเท่ากันของแต่ละค่าที่จะค้นหาขึ้นมาดำเนินการตรรกะ OR กันระหว่างบิตแมปเวกเตอร์

ตัวอย่างเช่น จากภาพประกอบ 3-4 ต้องการทราบว่า บนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 1,4 และ 5 ซึ่งแบ่งเป็นขั้นตอนเพื่อหาคำตอบได้ดังนี้

1.1) อ่านบิตแมปเวกเตอร์ I^1

1.2) อ่านบิตแมปเวกเตอร์ I^2

1.3) ดำเนินการตรรกะ NOT ของผลลัพธ์จากข้อ 1.2)

1.4) นำผลลัพธ์จากข้อ 1.1) และ 1.3) มาดำเนินการตรรกะ AND กันในระดับบิตต่อบิต เพื่อค้นหาข้อมูลค่า Y = 1

1.5) อ่านบิตแมปเวกเตอร์ I^4

1.6) อ่านบิตแมปเวกเตอร์ I^5

1.7) ดำเนินการตรรกะ NOT ของผลลัพธ์จากข้อ 1.6)

1.8) นำผลลัพธ์จากข้อ 1.5) และ 1.7) มาดำเนินการตรรกะ AND กันในระดับบิตต่อบิต เพื่อค้นหาข้อมูลค่า Y = 4

1.9) อ่านบิตแมปเวกเตอร์ I^5

1.10) อ่านบิตแมปเวกเตอร์ I^6

1.11) ดำเนินการตรรกะ NOT ของผลลัพธ์จากข้อ 1.10)

1.12) นำผลลัพธ์จากข้อ 1.9) และ 1.11) มาดำเนินการตรรกะ AND กันในระดับบิตต่อบิต เพื่อค้นหาข้อมูลค่า Y = 5

1.13) นำผลลัพธ์จากข้อ 1.4) 1.8) และ 1.12) มาดำเนินการตรรกะ OR ในระดับบิตต่อบิต ซึ่งคำตอบที่ได้คือ ผลลัพธ์ของบิตแมปเวกเตอร์ที่มีค่าบิตเท่ากับ 1 (ON) ในที่นี้คือ เรคอร์ดที่ 4, 6 และ 8

3.2.3 ข้อดีของดัชนีบิตแมปแบบช่วง

การทำดัชนีบิตแมปแบบช่วงใช้พื้นที่ในการจัดเก็บดัชนีน้อยกว่าการทำดัชนีบิตแมปแบบพื้นฐานมาก กล่าวคือสามารถลดพื้นที่ลงได้ครึ่งหนึ่ง นั้นคือ $\lceil C/2 \rceil$

3.2.4 ข้อจำกัดของดัชนีบิตแมปแบบช่วง

ไม่เหมาะกับแอทริบิวต์ที่มีค่าคาร์ดินอลิตี้สูง ๆ เพราะยังจัดว่าใช้พื้นที่มากอยู่ เมื่อเปรียบเทียบกับดัชนีแบบถัดไปที่จะกล่าวถึง คือ ดัชนีบิตแมปแบบเข้ารหัส

3.3 ดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index)

การทำดัชนีบิตแมปแบบเข้ารหัส [4] มีวิธีการสร้างดัชนีแบบไม่มีการบีบอัดดัชนีที่ใช้พื้นที่น้อยที่สุด ซึ่งในการแทนค่าของแอทริบิวต์ที่เลือกมาทำดัชนีบิตแมปแบบเข้ารหัสบนแอทริบิวต์ที่มีค่าคาร์ดินอร์ลิตี้เท่ากับ C ประกอบด้วย ประกอบด้วย $\lceil \log_2 C \rceil$ บิตแมปเวกเตอร์ นั้นคือ $A^{\lceil \log_2 c \rceil - 1}, ..., A^2, A^1, A^0$ และ ตารางการเทียบค่าการเข้ารหัส เพื่อกำหนดค่าให้กับแต่ละบิตแมปเวกเตอร์ บิตที่ i ของแต่ละบิตแมปเวกเตอร์ จะมีค่าเท่ากับ 0 หรือ 1 ขึ้นอยู่กับค่าของแอทริบิวต์เรคอร์ดที่ i ว่า เทียบค่าของแต่ละบิตแมปเวกเตอร์ตรงกับค่าใด นั่นหมายความว่าแต่ละค่าของแอทริบิวต์จะถูกแทนค่าด้วยทุกบิตแมปเวกเตอร์ พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบเข้ารหัสดังภาพประกอบ 3-5

RID	แอทริบิวต์ Y
1	3
2	2
3	3
4	1
5	13
6	4
7	6
8	5
9	14
10	3
11	4
12	2
13	3

(a) แอทริบิวต์ Y

$A^3A^2A^1A^0$
0011
0010
0011
0001
1101
0100
0110
0101
1110
0011
0100
0010
0011

(b) ตารางดัชนี

ค่าแอทริบิวต์ Y	$A^3A^2A^1A^0$
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110

(c) ตารางการเทียบค่าการเข้ารหัสบนแอทริบิวต์ Y

ภาพประกอบ 3-5 ดัชนีบิตแมปแบบเข้ารหัสบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 4 บิตแมปเวกเตอร์

จากภาพประกอบ 3-5 (b) แอทริบิวต์ Y มีคาร์ดินอลิตี้เท่ากับ 15 ดังนั้นการเข้ารหัสแต่ละค่าจะต้องใช้บิตแมปเวกเตอร์ เท่ากับ $\lceil \log_2 15 \rceil = 4$ ซึ่งรูปแบบการเข้ารหัส คือ $A^3A^2A^1A^0$ และค่าแต่ละบิตของบิตแมปเวกเตอร์จะใช้สัญลักษณ์ A^i แสดงค่าของตำแหน่งบิตที่ i ที่มีค่าเท่ากับ 1 และสัญลักษณ์ $\overline{A^i}$ แสดงค่าของตำแหน่งบิตที่ i ที่มีค่าเท่ากับ 0 เพื่อใช้เป็นฟังก์ชันในการเข้าถึงข้อมูล (Retrieval Boolean Function) เช่น แอทริบิวต์ที่มีค่าเป็น “3” มีการเข้ารหัสเป็น 0011 และฟังก์ชันในการเข้าถึงข้อมูลคือ $\overline{A^3}\,\overline{A^2}A^1A^0$

3.3.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบเข้ารหัส

สำหรับการค้นหาข้อมูลแบบค่าเท่ากับ (Equality Query) บนดัชนีบิตแมปแบบเข้ารหัส สามารถทำได้โดยการนำค่าที่ต้องการสอบถามไปเทียบค่ากับตารางการเทียบค่าว่าแต่ละบิตแมปเวกเตอร์เก็บค่าใด จากนั้นนำมาเทียบกับทุก ๆ เรคอร์ดบนตารางดัชนี ถ้าเรคอร์ดใดมีค่าที่ตรงกันกับค่าได้จากตารางการเทียบทุกบิตแมปเวกเตอร์ แล้วเรคอร์ดนั้นคือคำตอบ

ตัวอย่างเช่น จากภาพประกอบ 3-5 ต้องการทราบว่า บนแอทริบิวต์ Y มี เรคอร์ดใดบ้างที่มีค่าเท่ากับ 4 นั้นคือ Y = 4 สามารถทำได้โดยการอ่านค่าการเข้ารหัสจากตารางการเทียบค่า ซึ่งจะมีการเข้ารหัสเป็น 0100 หลังจากนั้นทำการอ่านบิตแมปเวกเตอร์ $A^3A^2A^1A^0$ เข้ามา เพื่อทำการค้นหาเรคอร์ดที่มี A^3, A^1 และ A^0 มีค่าเป็น 0 และ A^2 ที่มีค่าเป็น 1 ซึ่งผลลัพธ์ที่ได้ คือ เรคอร์ดที่ 6 และ 11 เป็นคำตอบ

3.3.2 การสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบเข้ารหัส

ในการสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบเข้ารหัส สามารถทำได้โดยการหาคำตอบของแต่ละสมาชิก แล้วนำมาดำเนินการตรรกะ OR (ในที่นี้ใช้สัญลักษณ์ + แทน OR) พิจารณา ตัวอย่าง เมื่อต้องการสอบถามข้อมูลแบบสมาชิก Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 1 4 และ 5 โดยใช้ตารางการเทียบค่าจากภาพประกอบ 3-5 จะได้ฟังก์ชันในการสอบถามข้อมูลเป็น $\overline{A^3}\,\overline{A^2}\,\overline{A^1}A^0 + \overline{A^3}A^2\overline{A^1}\,\overline{A^0} + \overline{A^3}A^2\overline{A^1}A^0$ จะเห็นว่าต้องตรวจสอบทุกบิตแมปเวกเตอร์ของแต่ละค่าจึงจะได้คำตอบของการสอบถาม

3.3.3 ข้อดีของดัชนีบิตแมปแบบเข้ารหัส

เมื่อพิจารณาการทำดัชนีแบบบิตแมปทุกวิธีที่ได้กล่าวมาข้างต้นแล้ว จะเห็นได้ว่าการทำดัชนีบิตแมปแบบเข้ารหัสมุ่งเน้นไปที่การใช้พื้นที่ให้มีประสิทธิภาพสูงสุด กล่าวคือการทำดัชนีบิตแมปแบบเข้ารหัสใช้พื้นที่น้อยที่สุด (จำนวนบิตแมปเวกเตอร์ที่ใช้มีค่าเท่ากับ $\lceil \log_2 C \rceil$ ของแอทริบิวต์ที่จะทำการสร้าง) ซึ่งส่งผลให้เราสามารถเก็บข้อมูลได้มากขึ้นหรือดำเนินการทำดัชนีบิตแมปกับแอทริบิวต์ที่มีคาร์ดินอลิตี้ที่สูงได้ นอกจากนั้นยังทำให้สามารถรับและส่งข้อมูลได้ครั้งละมาก ๆ จึงเสียค่าใช้จ่ายในการรับและส่งข้อมูลน้อยลง

3.3.4 ข้อจำกัดของดัชนีบิตแมปแบบเข้ารหัส

การทำดัชนีบิตแมปแบบเข้ารหัสจะมีจำนวนบิตแมปเวกเตอร์ที่ใช้มีค่าเท่ากับ $\lceil \log_2 C \rceil$ ของแอทริบิวต์ที่จะทำการสร้าง จะเห็นว่าดัชนีแบบนี้ไม่เหมาะสำหรับการสอบถามข้อมูลแบบค่าเท่ากัน เพราะต้องอ่านมาทุกบิตแมปเวกเตอร์ ($\lceil \log_2 C \rceil$) ทุก ๆ ครั้งที่มีการสอบถามข้อมูล โดยเฉพาะอย่างยิ่งแอทริบิวต์ที่มีค่าคาร์ดินอลิตี้สูง ๆ นอกจากนี้ดัชนีบิตแมปแบบเข้ารหัสไม่สามารถดำเนินการตรรกะระดับบิตระหว่างบิตแมปเวกเตอร์ของต่างแอทริบิวต์ได้

3.4 ดัชนีบิตแมปแบบกระจาย (Scatter Bitmap Index)

ดัชนีบิตแมปแบบกระจาย [3] คือ การใช้ 2 บิตแมปเวกเตอร์ในการแทนค่าแต่ละค่าของแอทริบิวต์ Y ที่เรานำมาทำดัชนี และใช้หนึ่งบิตแมปเวกเตอร์ในการแทนค่าหลายค่า (ดูภาพประกอบ 3-6 ประกอบ) ดังนั้นดัชนีบิตแมปแบบกระจายประกอบด้วย $2\lceil \sqrt{C} \rceil$ บิตแมปเวกเตอร์ แบ่งเป็น 2 กลุ่มใหญ่ คือ กลุ่ม บิตแมปเวกเตอร์ Z (มี $\left\lfloor \frac{C}{m-1} \right\rfloor + 1$ บิตแมปเวกเตอร์) และกลุ่มบิตแมปเวกเตอร์ L (มี $m-2$ บิตแมปเวกเตอร์) โดย ค่า m มีค่าเท่ากับ $\sqrt{C}+1$ โดยมีขั้นตอนการสร้างดัชนีบิตแมปแบบกระจาย

กำหนดให้ C คือ คาร์ดินอลิตี้ของแอทริบิวต์ที่นำมาสร้างดัชนี และ m คือ ดีกรีหรือจำนวนสมาชิกในแต่ละกลุ่ม Z

1. กำหนดลำดับที่ให้กับข้อมูลแต่ละค่าของแอทริบิวต์ที่จะนำมาสร้างดัชนี โดยเริ่มจาก 0, 1, 2, ..., $C-1$
2. คำนวนค่า m โดยใช้สูตร

$$\text{m} = \sqrt{C}+1$$

3. แยกค่าเรียงลำดับ (ที่ได้จากข้อ 1.) ไปยัง 2 กลุ่ม คือ กลุ่ม Z และกลุ่ม L ดังนี้

- การสร้างกลุ่ม Z

a) ให้ค่าลำดับที่แรกสุด (ลำดับที่ 0) อยู่ในกลุ่ม Z^0

b) กำหนดสมาชิกให้แก่แต่ละกลุ่ม Z ซึ่งใช้สัญลักษณ์ Z^j โดยใช้สูตร

$$\text{j} = \left\lfloor \frac{v}{m-1} \right\rfloor + 1$$

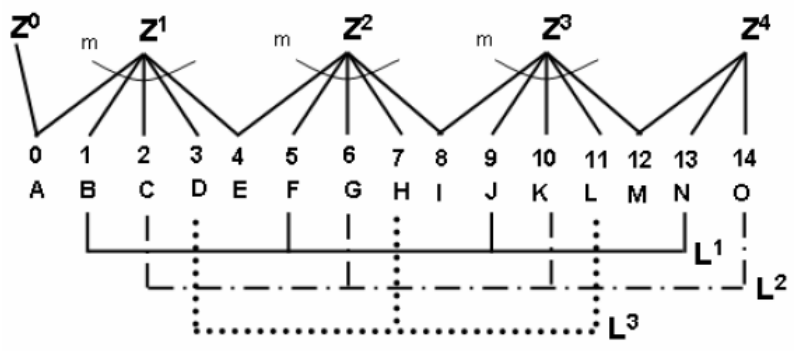
โดย v คือ ค่าลำดับที่ของข้อมูล (v = 0, 1, 2, ..., $C-1$) และให้ค่าลำดับที่หารด้วย m-1 ลงตัว อยู่ในกลุ่ม Z ทั้ง 2 กลุ่ม คือ กลุ่ม Z^{j-1} และ Z^j

- การสร้างกลุ่ม L

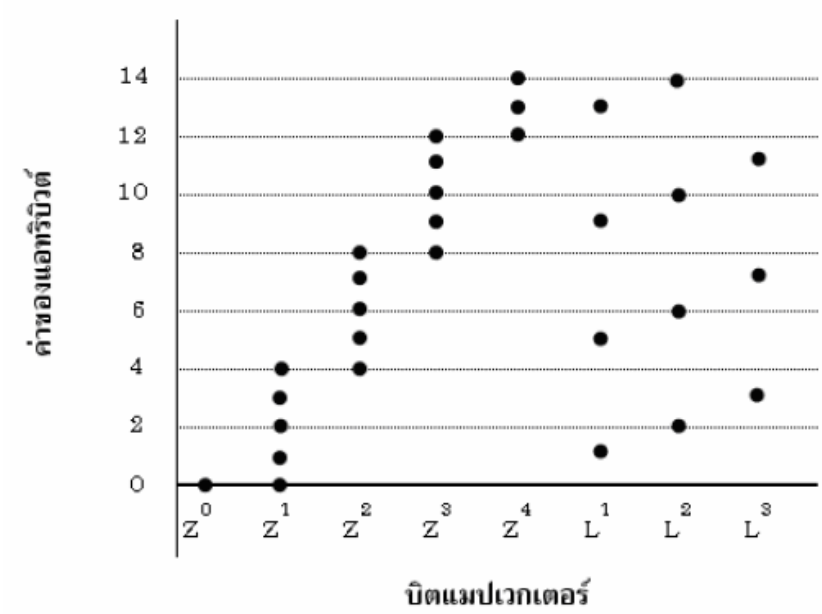
เป็นการกำหนดสมาชิกให้แก่แต่ละกลุ่ม L ซึ่งใช้สัญลักษณ์ L^k ยกเว้นค่าลำดับที่ k = 0 โดยใช้สูตร

$$\text{k} = \text{v MOD (m-1)}$$

โดย v คือ ค่าลำดับที่ของข้อมูล (v = 0, 1, 2, 3, ..., $C-1$)



ภาพประกอบ 3-6 รูปแบบทั่วไปของดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ Y (มีค่า C = 15)โดยมีจำนวนสมาชิกภายในกลุ่ม Z $\leq$ 5 (m = 5)



ภาพประกอบ 3-7 แผนภาพการลงรหัสดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ Y เมื่อ $C = 15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-6 แอทริบิวต์ Y มีค่าที่เป็นไปได้ 15 ค่า คือ 0,1,2,...,14 ดังนั้น ดัชนีบิตแมปแบบกระจายประกอบด้วย $8\left(=\left\lceil 2\sqrt{15}\right\rceil\right)$ บิตแมปเวกเตอร์ คือ กลุ่ม Z มี $5\left(=\lceil 15/(5-1)\rceil+1,\ m=\left\lceil\sqrt{15}\right\rceil+1=5\right)$ บิตแมปเวกเตอร์ คือ Z^0, Z^1, Z^2, Z^3, Z^4 และกลุ่ม L มี $3\,(=5-2)$ บิตแมปเวกเตอร์ คือ L^1, L^2, L^3 พิจารณาตัวอย่างดัชนีบิตแมปแบบกระจายดังภาพประกอบ 3-7

RID	แอทริบิวต์ Y	L^3	L^2	L^1	Z^4	Z^3	Z^2	Z^1	Z^0
1	H	1	0	0	0	0	1	0	0
2	D	1	0	0	0	0	0	1	0
3	L	1	0	0	0	1	0	0	0
4	C	0	1	0	0	0	0	1	0
5	D	1	0	0	0	0	0	1	0
6	B	0	0	1	0	0	0	1	0
7	N	0	0	1	1	0	0	0	0
8	A	0	0	0	0	0	0	1	1
9	G	0	1	0	0	0	1	0	0
:	:	:	:	:	:	:	:	:	:
100	F	0	0	1	0	0	1	0	0

ภาพประกอบ 3-8 ตัวอย่างการทำดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ Y (มีค่า C=15, m=5) ใช้ 8 บิตแมปเวกเตอร์

ตัวอย่างการทำดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ Y จากภาพประกอบ 3-8 เรคอร์ดที่ 5 และ 2 ของแอทริบิวต์ Y มีค่าเท่ากับ D กล่าวคือ จากภาพประกอบ 3-6 D มีค่าเท่ากับ 3

<u>สร้างบิตแมปเวกเตอร์กลุ่ม Z</u>

$$j = \left\lfloor \frac{v}{m-1} \right\rfloor + 1 = \left\lfloor \frac{3}{5-1} \right\rfloor + 1 = 1$$

ดังนั้น บิตที่ 1 ของบิตแมปเวกเตอร์ Z^1 มีค่าเท่ากับ 1

<u>สร้างบิตแมปเวกเตอร์กลุ่ม L</u>

$$k = v \bmod (m-1) = 3 \bmod (5-1) = 3$$

ดังนั้น บิตที่ 1 ของบิตแมปเวกเตอร์ L^3 มีค่าเท่ากับ 1

3.4.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบกระจาย

การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบกระจายมีรูปแบบทั่วไปคือ

$$"Y = v =" \begin{cases} Z^{v/(m-1)} \wedge Z^{v/(m-1)+1} & \text{ถ้า } v \mod(m-1) = 0 \\ Z^{v/(m-1)+1} \wedge L^{v \bmod (m-1)} & \text{กรณีอื่น ๆ} \end{cases}$$

ตัวอย่างเช่น จากภาพประกอบ 3-8 ต้องการทราบว่าบนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ D กล่าวคือ จากภาพประกอบ 3-6 D มีค่าเท่ากับ 3 ซึ่งตรงกับเงื่อนไขที่ $2\,(3 \bmod (5-1) \neq 0)$ ดังนั้น คำตอบสามารถหาได้จาก

$$\begin{aligned} "Y = 3" &= Z^{v/(m-1)+1} \wedge L^{v \bmod (m-1)} \\ &= Z^{3/(5-1)+1} \wedge L^{3 \bmod (5-1)} \\ &= Z^1 \wedge L^3 \end{aligned}$$

ซึ่งแบ่งเป็นขั้นตอนเพื่อหาคำตอบได้ดังนี้

1) อ่านบิตแมปเวกเตอร์ Z^1

2) อ่านบิตแมปเวกเตอร์ L^3

3) ดำเนินการตรรกะ AND บิตต่อบิตระหว่างบิตแมปเวกเตอร์ที่ได้จากข้อ 1) และ 2)

4) คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้จากข้อ 3 พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น 3 คือเรคอร์ดที่ 2 และ 5

3.4.2 การสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบกระจาย

ในการสอบถามแบบสมาชิกบนดัชนีบิตแมปแบบกระจาย สามารถทำได้โดยการหาคำตอบของแต่ละสมาชิก แล้วนำมาดำเนินการตรรกะ OR

ตัวอย่างเช่น จากภาพประกอบ 3-8 ถ้าต้องการค้นหา Y = B, G และ H กล่าวคือ จากภาพประกอบ 3-6 จำได้ว่า Y=1 6 และ 7

<u>วิธีการดึงข้อมูล</u>

1) ตรวจสอบดูว่า Y=1 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{1}{5-1} \right\rfloor + 1 = 1$

2) ตรวจสอบดูว่า Y=1 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 1 mod (5-1) = 1

3) จะได้ว่า Y=1 อยู่ในกลุ่ม Z^1 และ L^1 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^1 ดำเนินการตรรกะ AND กับ L^1 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^1 \wedge L^1$ ที่มีค่าบิตเท่ากับ 1

4) ตรวจสอบดูว่า Y=6 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{6}{5-1} \right\rfloor + 1 = 2$

5) ตรวจสอบดูว่า Y=6 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 6 mod (5-1) = 2

6) จะได้ว่า Y=6 อยู่ในกลุ่ม Z^2 และ L^2 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^2 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^2$ ที่มีค่าบิตเท่ากับ 1

7) ตรวจสอบดูว่า Y=7 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{7}{5-1} \right\rfloor + 1 = 2$

8) ตรวจสอบดูว่า Y=7 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 7 mod (5-1) = 3

9) จะได้ว่า Y=7 อยู่ในกลุ่ม Z^2 และ L^3 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^3 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^3$ ที่มีค่าบิตเท่ากับ 1

10) นำผลที่ได้มา OR กัน จะได้ว่า $\left(Z^1 \wedge L^1\right) \vee \left(Z^2 \wedge L^2\right) \vee \left(Z^2 \wedge L^3\right)$ คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้ พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น B G และ H คือเรคอร์ดที่ 1 6 และ 9

3.4.3 ข้อดีของดัชนีบิตแมปแบบกระจาย

ในกรณีที่พิจารณาเรื่องการใช้พื้นที่ของดัชนีบิตแมปแบบกระจาย พบว่าดัชนีบิตแมปแบบกระจาย ประหยัดพื้นที่ได้มากกว่าดัชนีบิตแมปแบบพื้นฐาน และ ดัชนีบิตแมปแบบช่วง

ในกรณีที่พิจารณาเรื่องการสอบถามข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิกของดัชนีบิตแมปแบบกระจาย พบว่าจะใช้เวลาในการสอบถามข้อมูลน้อยกว่า ดัชนีบิตแมปแบบเข้ารหัส และ แบบช่วง เพราะบิตแมปเวกเตอร์ที่ถูกอ่าน มีจำนวน 2 บิตแมปเวกเตอร์ และในการค้นหาค่าเท่ากันมีการดำเนินการตรรกะ AND เกิดขึ้นเพียงตัวดำเนินการเดียวเท่านั้น

3.4.4 ข้อจำกัดของดัชนีบิตแมปแบบกระจาย

ดัชนีบิตแมปแบบกระจาย จะใช้พื้นที่ในการจัดเก็บดัชนีมากกว่าดัชนีบิตแมปแบบเข้ารหัส แต่ดัชนีบิตแมปแบบกระจายจะค้นหาได้เร็วกว่าดัชนีบิตแมปแบบเข้ารหัส และดัชนีบิตแมปแบบกระจายจะใช้พื้นที่มากกว่าดัชนีบิตแมปแบบช่วง ในกรณีที่แอทริบิวต์ที่นำมาสร้างดัชนีมีค่าคาร์ดินอลิตี้น้อยกว่า 15

อย่างไรก็ตาม ในภาพรวม ดัชนีบิตแมปแบบกระจายจะมีประสิทธิภาพและเหมาะกับการสอบถามข้อมูลแบบค่าเท่ากันและแบบความเป็นสมาชิกมากที่สุด เมื่อเปรียบเทียบกับดัชนีบิตแมปแบบพื้นฐาน ดัชนีบิตแมปแบบช่วง และดัชนีบิตแมปแบบเข้ารหัส ดังตาราง 3-1

ตาราง 3-1 ลักษณะที่สำคัญของดัชนีบิตแมปแบบพื้นฐาน แบบช่วง แบบเข้ารหัส และแบบกระจาย (C คือ คาร์ดินอลิตี้ของแอทริบิวต์ที่เลือกมาทำดัชนี)

ชนิดของดัชนีบิตแมป	พื้นที่	เวลา	
	จำนวนบิตแมปเวกเตอร์	การสอบถามข้อมูล	
		แบบเท่ากัน	แบบสมาชิก
แบบพื้นฐาน [2]	C	1:0	n: n-1 (OR)
แบบช่วง [2]	$\left\lceil \frac{C}{2} \right\rceil$	2:2 (1AND,1NOT)	2 n : 3 n -1(n AND, n-1OR, n NOT)
แบบเข้ารหัส [4]	$\lceil \log_2 C \rceil$	$\lceil \log_2 C \rceil$: ใช้ mapping table	n $\lceil \log_2 C \rceil$: ใช้ mapping table, n-1 OR
แบบกระจาย [3]	$2\lceil \sqrt{C} \rceil$	2:1 (1AND)	2n:2n-1(n AND, n-1OR)

บทที่ 4

การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

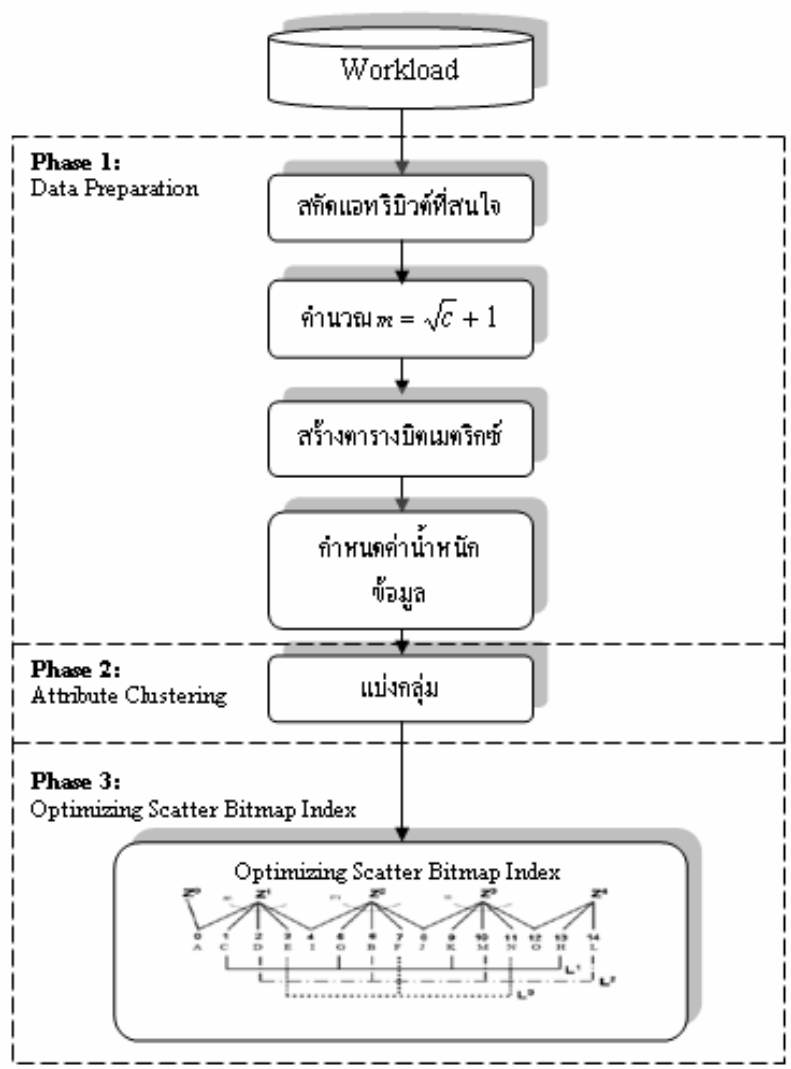
จากบทที่ 3 ได้อธิบายเกี่ยวกับเทคนิคที่เกี่ยวกับเทคนิคการทำดัชนีแบบบิตแมปแบบต่าง ๆในคลังข้อมูล ซึ่งงานวิจัยแต่ละชิ้นมุ่งเน้นในการลดพื้นที่ในการจัดเก็บ และเพิ่มความเร็วในการค้นหาข้อมูล หรือทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล และได้มีงานวิจัยที่ยังแบ่งประเภทของการค้นหาข้อมูล (Query) เข้ามาเกี่ยวข้อง ซึ่งแยกเป็นการค้นหาแบบเท่ากันและค้นหาแบบสมาชิก เช่น ดัชนีบิตแมปแบบพื้นฐาน [2] เป็นดัชนีที่มีความเร็วในการค้นหาข้อมูลแบบเท่ากัน แต่ไม่เหมาะสำหรับแอทริบิวต์สูง ๆ เพราะใช้พื้นที่จัดเก็บมาก ดัชนีบิตแมปแบบเข้ารหัส [4] ซึ่งเป็นดัชนีบิตแมปที่เน้นการลดพื้นที่การจัดเก็บแต่ใช้เวลานานในการค้นหาข้อมูล ดัชนีบิตแมปแบบกระจาย [3,11] จะรวมทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล เป็นต้น นอกจากนี้ยังพบอีกว่าจากโครงสร้างของดัชนีบิตแมปแบบกระจาย [3] ยังสามารถเพิ่มความเร็วในการค้นหาข้อมูลได้อีก โดยการพิจารณาการค้นหาข้อมูลที่อยู่ในกลุ่มสมาชิกเดียวกัน

งานวิจัยนี้ ได้เสนอเทคนิคการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิก ของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้น โดยใช้การแบ่งกลุ่มข้อมูล (Data Clustering) [5,6,7,9,10,17] ซึ่งเป็นหนึ่งในเทคนิคการทำเหมืองข้อมูล (Data Mining) [5,6,7,10,11,13,17] มาช่วยในการจัดกลุ่มและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจาย [3,11]

ในบทนี้เป็นการนำเสนอการเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูล (Scatter Bitmap Index Optimization using Data Clustering : SBIOC) มาช่วยในการจัดกลุ่มข้อมูลและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลในคลังข้อมูล แต่ละส่วนมีรายละเอียดและหลักการที่จะกล่าวต่อไป

4.1 การออกแบบ การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ส่วนนี้เป็นการนำเสนอการเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูล (Scatter Bitmap Index Optimization using Data Clustering: SBIOC) มาช่วยในการจัดกลุ่มข้อมูลและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลในคลังข้อมูล ภาพประกอบ 4-1 แสดงโครงสร้างของ SBIOC โดยแบ่งขั้นตอนการทำงานออกเป็น 3 ส่วนหลักคือ การเตรียมข้อมูล การแบ่งกลุ่ม และการเพิ่มประสิทธิภาพ ดัชนีบิตแมปแบบกระจาย แต่ละส่วนมีรายละเอียดดังนี้



ภาพประกอบ 4-1 ขั้นตอนการทำงานของการเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (SBIOC)

- **ขั้นตอนที่ 1: การเตรียมข้อมูล (Data Preparation)**

การเตรียมข้อมูลประกอบด้วย 4 ขั้นตอนดังนี้

1. สกัดแอทริบิวต์ที่สนใจจะสร้างดัชนีบิตแมปจาก workload เพื่อสร้างตารางค่าแอทริบิวต์

2. คำนวณหาค่าจำนวนสมาชิกในแต่ละกลุ่ม(m) จากสมการ m = $\sqrt{C}+1$ โดยค่า C คือค่าที่เป็นไปได้ของแอทริบิวต์ที่ถูกสร้างดัชนีบิตแมป

3. สร้างตารางบิตเมตริกซ์จากตารางค่าแอทริบิวต์

4. ทำการกำหนดค่าน้ำหนักให้แก่แต่ละ item (แต่ละค่าที่ปรากฏในการสอบถามแบบสมาชิกในแต่ละ query) และทำการรวมค่าน้ำหนักของทุก query ในแต่ละ item

คุณลักษณะพื้นฐาน ของการทำดัชนีบิตแมปแบบกระจาย คือ การใช้สองบิตแมปเวกเตอร์ในการแทนค่าหนึ่งค่า และการใช้หนึ่งบิตแมปเวกเตอร์ในการแทนค่าหลายค่า (ในที่นี้คือค่า m รายละเอียดการหาค่า m สามารถดูได้จากอ้างอิง [3]) ดังนั้นในการกำหนดค่าน้ำหนักให้แต่ละ item ของแต่ละ query ใด ๆ (Q ใด ๆ) สามารถพิจารณาได้ 3 กรณีดังนี้ (ดูตาราง 4-1)

1. ถ้าผลรวมของสมาชิก item ที่ปรากฏใน query ใด ๆ มีค่าเท่ากับ 1 (การค้นหาแบบเท่ากัน) หรือไม่ปรากฏใน query ใด ๆ แล้วจะให้ค่าน้ำหนักแก่ Item นั้นเป็น 0

2. ถ้าผลรวมของสมาชิก item ที่ปรากฏใน query ใด ๆ มีค่าไม่เท่ากับจำนวนเท่าของ m (เช่น 2, 3 หรือ rm โดย r = 1, 2, 3,..) ไม่เท่ากับ 1 และ ไม่เท่ากับ 0 แล้วจะให้ค่าน้ำหนักแก่แต่ละ Item เหล่านั้นเป็น 2

3. ถ้าผลรวมของสมาชิก Item ที่ปรากฏใน query ใดๆ มีค่าเท่ากับจำนวนเท่าของ m (rm โดย r = 1, 2, 3, ..) แล้วจะให้ค่าน้ำหนักแก่แต่ละ item เหล่านั้นเป็น 4

ตาราง 4-1 การกำหนดค่าน้ำหนักของแต่ละ item

น้ำหนักของ Item	คำอธิบาย
0	สมาชิก item ที่ปรากฏใน query ใดๆ มี 1 item หรือ ไม่ปรากฏใน query
2	สมาชิก item ที่ปรากฏใน query ใดๆ เมื่อรวมกันไม่เท่ากับจำนวนเท่าของ m ไม่เท่ากับ 1 และ ไม่เท่ากับ 0
4	สมาชิก item ที่ปรากฏใน query ใดๆ เมื่อรวมกันเท่ากับจำนวนเท่าของ m (rm เมื่อ r=1, 2, 3 ..)

- **ขั้นตอนที่ 2: การแบ่งกลุ่ม (Attribute Clustering)**

การแบ่งกลุ่ม เป็นการจัดกลุ่มค่าของแอทริบิวต์ที่คล้ายกันมีน้ำหนักใกล้เคียงกันอยู่ในกลุ่มเดียวกัน โดยงานวิจัยชิ้นนี้ได้เลือกใช้วิธีการ K-Mode Clustering เนื่องจากวิธีดังกล่าวเหมาะกับ SBIOC ซึ่งแอทริบิวต์ที่นำมาสร้างดัชนีมักเป็นข้อมูลที่ไม่สามารถนำมาคำนวณได้ โดยกำหนดจำนวนกลุ่ม K = m -1 กลุ่มและแต่ละกลุ่มมีสมาชิกไม่เกินค่า m

- **ขั้นตอนที่ 3: การเพิ่มประสิทธิภาพดัชนีบิตแมปแบบกระจาย (Optimizing Scatter Bitmap Index)**

ในส่วนนี้ เป็นการสร้างดัชนีบิตแมปแบบกระจายตามที่ได้อธิบายไว้ใน [3] โดยนำผลลัพธ์ที่ได้จากการแบ่งกลุ่มของค่าของแอทริบิวต์มาใช้ในการลงรหัส ผลลัพธ์ที่ได้ในส่วนนี้คือรูปแบบการลงรหัสของดัชนีบิตแมปแบบกระจายที่ทำให้ประสิทธิภาพในการสอบถามดีขึ้น

ข้อมูลในภาพประกอบ 4-2 จะถูกใช้ในการอธิบายการทำงานของ SBIOC โดยภาพประกอบ 4-2(a) แสดงตัวอย่าง workload และ ภาพประกอบ 4-2(b) แสดงตารางค่าแอทริบิวต์ Y ที่สกัดจาก workload

Q1: SELECT * FROM T WHERE Y in (A, B, C, E, J, K, M, N)
Q2: SELECT * FROM T WHERE Y in (D, F, H, L, N)
Q3: SELECT * FROM T WHERE Y in (G, J)
Q4: SELECT * FROM T WHERE Y in (A, B, D, E, F, H, K, L, M, N)

(a) ตัวอย่าง workload

Query	ค่าแอทริบิวต์ Y
Q1	A, B, C, E, J, K, M, N
Q2	D, F, H, L, N
Q3	G, J
Q4	A, B, D, E, F, H, K, L, M, N

(b) ตารางค่าแอทริบิวต์ Y

ภาพประกอบ 4-2 ตารางค่าแอทริบิวต์ Y ที่สกัดจาก workload

- **ขั้นตอนที่ 1: การเตรียมข้อมูล (Data Preparation)**

การเตรียมข้อมูลประกอบด้วย 4 ขั้นตอนดังนี้

1. สกัดแอทริบิวต์ที่สนใจที่จะสร้างดัชนีบิตแมป จาก workload เพื่อสร้างตารางค่าแอทริบิวต์ จากตัวอย่าง workload ดังภาพประกอบ 4-2(a) พบว่า แอทริบิวต์ที่น่าสนใจ คือ แอทริบิวต์ Y เนื่องจากตัวอย่าง workload จะมีการสอบถามข้อมูลจากแอทริบิวต์ Y ทุกตัว

2. คำนวณหาค่าจำนวนสมาชิกในแต่ละกลุ่ม (m) จากสมการ $m = \sqrt{C} + 1$ โดยค่า C คือค่าที่เป็นไปได้ของแอทริบิวต์ที่ถูกสร้างดัชนีบิตแมป จากตัวอย่าง workload ดังภาพประกอบ 4-2(a) จะพบว่า C = 15 ซึ่งจะได้ว่า

$$\begin{aligned} m &= \sqrt{C} + 1 \\ &= \sqrt{15} + 1 \\ &= 5 \end{aligned}$$

ดังนั้น จากตัวอย่าง workload ดังภาพประกอบ 4-2(a) จะได้ค่า m = 5

3. สร้างตารางบิตเมตริกซ์จากตารางค่าแอทริบิวต์ โดยกำหนดให้บิตเมตริกซ์เป็น bitMatrix[i,j] เมื่อ i เป็นเลขลำดับของการค้นหาข้อมูลใน workload และ j เป็นเลขตำแหน่งลำดับค่าในแอทริบิวต์ จากตัวอย่าง workload ดังภาพประกอบ 4-2(a) ค่า B ในการสอบถามข้อมูลที่ 4 (Q4) จะได้ว่า bitMatrix[4,2] จะมีค่าเท่ากับ 1 ดังภาพประกอบ 4-3(a)

4. ทำการกำหนดค่าน้ำหนักให้แก่แต่ละ item (แต่ละค่าที่ปรากฏในการสอบถามแบบสมาชิกในแต่ละ query) โดยการให้น้ำหนักจะพิจารณาจากตาราง 4-1 ประกอบ จากตัวอย่าง workload ดังภาพประกอบ 4-2(b) Q1 มีสมาชิก item รวมกันแล้วจะได้เท่ากับ 8 ซึ่งไม่เท่ากับจำนวนเท่าของ m (นั้นคือมีค่าเท่ากับ m+3) ดังนั้น น้ำหนักของสมาชิก item ดังกล่าวทุก item มีค่าเท่ากับ 2 (ดังที่ได้กล่าวไว้ในตาราง 4-1) ตัวอย่างเช่น ค่า A ในการสอบถามที่ 1 (Q1) มีการกำหนดบิตเมตริกซ์เป็น bitMatrix[1,1] จะมีค่าน้ำหนักเป็น 2 ในตำแหน่งลำดับที่ 1 ของบิตเมตริกซ์ Q1 เนื่องจาก การสอบถามที่ 1 (Q1) item ทุกตัวจะมีค่าเท่ากับ 2 เป็นต้น ในทำนองเดียวกัน Q2 จะมีสมาชิก item รวมกันแล้วจะได้เท่ากับ 5 ซึ่งเท่ากับจำนวนเท่าของ m (นั้นคือมีค่าเท่ากับ rm เมื่อ r = 1, 2,..) ดังนั้น น้ำหนักของสมาชิก item ดังกล่าวทุก item มีค่าเท่ากับ 4 ในทำนองเดียวกัน Q3 จะมีสมาชิก item รวมกันแล้วจะได้เท่ากับ 2 ซึ่งไม่เท่ากับจำนวนเท่าของ m (นั้นคือมีค่าเท่ากับ m-3) ดังนั้น น้ำหนักของสมาชิก item ดังกล่าวทุก item มีค่าเท่ากับ 2 ในทำนองเดียวกัน Q4 จะมีสมาชิก item รวมกันแล้วจะได้เท่ากับ 10 ซึ่งเท่ากับจำนวนเท่าของ m (นั้นคือมีค่าเท่ากับ rm เมื่อ r = 1, 2,..) ดังนั้น น้ำหนักของสมาชิก item ดังกล่าวทุก item มีค่าเท่ากับ 4

ภาพประกอบ 4-3(b) แสดงตัวอย่าง ตารางบิตเมตริกซ์และกำหนดน้ำหนักข้อมูล

Query	ค่าแอทริบิวต์ Y														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q1	1	1	1	0	1	0	0	0	0	1	1	0	1	1	0
Q2	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0
Q3	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Q4	1	1	0	1	1	1	0	1	0	0	1	1	1	1	0

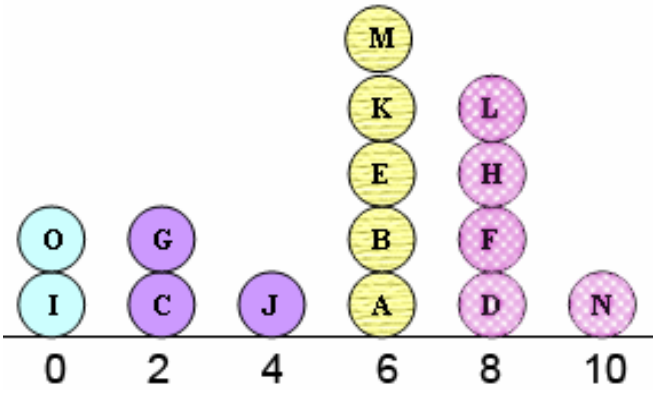
(a) ตารางบิตเมตริกซ์

Query	ค่าแอทริบิวต์ Y														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q1	2	2	2	0	2	0	0	0	0	2	2	0	2	2	0
Q2	0	0	0	4	0	4	0	4	0	0	0	4	0	4	0
Q3	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0
Q4	4	4	0	4	4	4	0	4	0	0	4	4	4	4	0

(b) ตารางบิตเมตริกซ์หลังจากกำหนดน้ำหนักข้อมูล

ค่าแอทริบิวต์ Y	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
น้ำหนักรวม	6	6	2	8	6	8	2	8	0	4	6	8	6	10	0

(c) รวมน้ำหนักของข้อมูลก่อนการแบ่งกลุ่มข้อมูล



(d) ผลจากการแบ่งกลุ่มข้อมูล

ภาพประกอบ 4-3 ตัวอย่างตารางการให้น้ำหนักและผลจากการแบ่งกลุ่มข้อมูล

หลังจากนั้นทำการทำการรวมค่าน้ำหนักของทุก การสอบถาม (query) ในแต่ละ item เช่น item A ได้ค่าน้ำหนักรวมเป็น 6 เป็นต้น ภาพประกอบ 4-3(c) แสดงตารางการรวมค่าน้ำหนักของทุกค่าที่เป็นไปได้ของแอทริบิวต์ Y

- **ขั้นตอนที่ 2: การแบ่งกลุ่ม (Attribute Clustering)**

ในส่วนนี้จะกล่าวถึงการแบ่งกลุ่มข้อมูล ก่อนที่จะทำการแบ่งกลุ่มข้อมูล จะต้องทำการจัดกลุ่มข้อมูลหลังจากการให้น้ำหนักข้อมูลแล้ว โดยนำข้อมูลที่มีค่าน้ำหนักรวมเท่ากันจัดให้อยู่ในแถวเดียวกัน ดังภาพประกอบ 4-3(d) ตัวอย่างการจัดกลุ่มข้อมูล (ดูภาพประกอบ 4-3 (c) ประกอบ)

1. ข้อมูล O และ I ซึ่งมีค่าน้ำหนักรวมเป็น 0 จัดให้อยู่ในแถวเดียวกันในแถวที่มีค่าเท่ากับ 0

2. ข้อมูล G และ C ซึ่งมีค่าน้ำหนักรวมเป็น 2 จัดให้อยู่ในแถวเดียวกันในแถวที่มีค่าเท่ากับ 2

3. ข้อมูล J ซึ่งมีค่าน้ำหนักรวมเป็น 4 จัดให้อยู่ในแถวที่มีค่าเท่ากับ 4 ซึ่งในที่นี้มีเพียงข้อมูลเดียว

4. ข้อมูล M K E B และ A ซึ่งมีค่าน้ำหนักรวมเป็น 6 จัดให้อยู่ในแถวเดียวกัน กล่าวคือ ในแถวที่มีค่าเท่ากับ 6

5. ข้อมูล L H F และ D ซึ่งมีค่าน้ำหนักรวมเป็น 8 จัดให้อยู่ในแถวเดียวกันในแถวที่มีค่าเท่ากับ 8

6. ข้อมูล N ซึ่งมีค่าน้ำหนักรวมเป็น 10 จัดให้อยู่ในแถวที่มีค่าเท่ากับ 10 ซึ่งในที่นี้มีเพียงข้อมูลเดียว

จากนั้นทำการแบ่งกลุ่มข้อมูล โดยการแบ่งกลุ่ม เป็นการจัดกลุ่มสมาชิกของแอทริบิวต์ซึ่งได้กำหนดน้ำหนักแล้ว โดยใช้เทคนิคของอัลกอริทึม K-Mode (ตามที่ได้กล่าวไว้ในหัวข้อ 2.5) โดยจะกล่าวในรายละเอียดดังนี้

1.ทำการหาข้อมูล K ตัวซึ่งใช้เป็นค่าศูนย์กลางหรือตัวแทนกลุ่มในแต่ละกลุ่มของ K กลุ่ม ซึ่งในที่นี้จะกำหนดจำนวนกลุ่มจากสมการ K = m -1 นั่นคือ 5-1 = 4 กลุ่ม และ 4 กลุ่มดังกล่าวจะเลือกตามหลักการของการเลือกค่าศูนย์กลางหรือตัวแทนกลุ่ม K-Mode (ภาพประกอบ 2-3) ซึ่งมีดังนี้ กลุ่มที่ 1 เป็นกลุ่มที่มีค่าน้ำหนักเป็น 0 นั้นคือมี I และ O กลุ่มที่ 2 เป็นกลุ่มที่มีค่าน้ำหนักเป็น 2 นั้นคือมี C และ G กลุ่มที่ 3 เป็นกลุ่มที่มีค่าน้ำหนักเป็น 6 นั้นคือมี A B E K และ M กลุ่มที่ 4 เป็นกลุ่มที่มีค่าน้ำหนักเป็น 8 นั้นคือมี D F H และ L

2.ทำการอ่านข้อมูล จัดให้ข้อมูลแต่ละตัวอยู่ในกลุ่มที่มีค่าระยะห่างกันน้อยที่สุด (เหมือนกันมากที่สุด) นั้นคือ นำ J ซึ่งมีค่าน้ำหนักเท่ากับ 4 นำมาเข้ากลุ่มสมาชิกที่มีค่าน้ำหนักเป็น 2 จะได้ว่ากลุ่ม 2 จะมีสมาชิก C G และ J ในทำนองเดียวกัน นั้นคือ นำ N ซึ่งมีค่าน้ำหนัก

เท่ากับ 10 นำมาเข้ากลุ่มสมาชิกที่มีค่าน้ำหนักเป็น 8 จะได้ว่ากลุ่ม 4 จะมีสมาชิก D F H L และ N

3. หาค่าศูนย์กลางกลุ่มของแต่ละกลุ่มใหม่ จำนวน 4 กลุ่ม นั้นคือจะได้กลุ่มเดิม

4. ทำการหากลุ่มให้ข้อมูลทุกตัวจากศูนย์กลางใหม่ที่ได้ นั้นคือจะได้กลุ่มเดิม

5.ทำซ้ำในขั้นตอนที่ 3 และ 4 จนกว่าข้อมูลทุกตัวจะไม่มีการเปลี่ยนกลุ่ม

ภาพประกอบ 4-3 (d) แสดงค่าที่เป็นไปได้ของแอทริบิวต์ Y ที่ผ่านการให้ค่าน้ำหนักและทำการจัดเรียงตามค่าน้ำหนักพร้อมที่จะทำการจัดกลุ่ม จากการใช้อัลกอริทึม K-Mode ทำการจัดกลุ่ม 4 กลุ่มมีรายละเอียดดังนี้ กลุ่มที่ 1 คือ D F H L N กลุ่มที่ 2 คือ A B E K M กลุ่มที่ 3 คือ C G J กลุ่มที่ 4 คือ I O โดยพิจารณาจากระยะทาง

- **ขั้นตอนที่ 3: การเพิ่มประสิทธิภาพดัชนีบิตแมปแบบกระจาย (Optimizing Scatter Bitmap Index)**

ในส่วนนี้จะกล่าวถึงการเพิ่มประสิทธิภาพดัชนีบิตแมปแบบกระจาย ซึ่งนั้นคือการนำผลของการแบ่งกลุ่มข้อมูลนำมาสร้างเป็นดัชนีบิตแมปแบบกระจาย ซึ่งมีรายละเอียดดังนี้

1. นำสมาชิกที่ได้แบ่งกลุ่มไปแล้วใน ขั้นตอนที่ 2 นั้นคือ กลุ่มที่ 1 คือ D F H L N กลุ่มที่ 2 คือ A B E K M กลุ่มที่ 3 คือ C G J กลุ่มที่ 4 คือ I O นำมาจัดวางตามโครงสร้างดัชนีบิตแมปแบบกระจาย จะได้ ดังภาพประกอบ 4-4

2. กำหนดลำดับที่ให้กับข้อมูลแต่ละค่าของแอทริบิวต์ที่จะนำมาสร้างดัชนี โดยเริ่มจาก 0, 1, 2, ..., $C-1$

3. แยกค่าเรียงลำดับ (ที่ได้จากข้อ 1.) ไปยัง 2 กลุ่ม คือ กลุ่ม Z และกลุ่ม L ดังนี้

การสร้างกลุ่ม Z

- จำนวนบิตแมปเวกเตอร์ Z ที่ต้องสร้าง = $\left\lceil \frac{C}{m-1} \right\rceil + 1 = \left\lceil \frac{15}{5-1} \right\rceil + 1$

$= \left\lceil \frac{15}{3} \right\rceil + 1 = 5$ ซึ่งก็คือ บิตแมปเวกเตอร์ Z^0 ... Z^4 โดยกำหนดให้

- ให้ค่าลำดับที่แรกสุด (ลำดับที่ 0) อยู่ในกลุ่ม Z^0

- บิตแมปเวกเตอร์ Z^1 แทนค่า 0, 1, 2, 3

เพราะ $j = \left\lfloor \frac{v}{m-1} \right\rfloor + 1 = 1$ เมื่อ v = 0, 1, 2, 3

และ 4 ถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^2 เพราะ 4 หารด้วย m-1 ลงตัว ซึ่งในที่นี้ m-1 มีค่าเท่ากับ 4 ดังนั้น 4 จึงถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^{j-1} (ในที่นี้ $Z^{j-1} = Z^{2-1}$ ซึ่งก็คือ Z^1)

- บิตแมปเวกเตอร์ Z^2 แทนค่า 4, 5, 6, 7

เพราะ $j = \left\lfloor \frac{v}{m-1} \right\rfloor + 1 = 2$ เมื่อ v = 4, 5, 6, 7

และ 8 ถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^3 เพราะ 8 หารด้วย m-1 ลงตัว ซึ่งในที่นี้ m-1 มีค่าเท่ากับ 4 ดังนั้น 8 จึงถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^{j-1} (ในที่นี้ $Z^{j-1} = Z^{3-1}$ ซึ่งก็คือ Z^2)

- บิตแมปเวกเตอร์ Z^3 แทนค่า 8, 9, 10, 11

เพราะ $j = \left\lfloor \frac{v}{m-1} \right\rfloor + 1 = 3$ เมื่อ v = 8, 9, 10, 11

และ 12 ถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^4 เพราะ 12 หารด้วย m-1 ลงตัว ซึ่งในที่นี้ m-1 มีค่าเท่ากับ 4 ดังนั้น 12 จึงถูกแทนค่าด้วยบิตแมปเวกเตอร์ Z^{j-1} (ในที่นี้ $Z^{j-1} = Z^{4-1}$ ซึ่งก็คือ Z^3)

- บิตแมปเวกเตอร์ Z^4 แทนค่า 12, 13, 14

เพราะ $j = \left\lfloor \frac{v}{m-1} \right\rfloor + 1 = 4$ เมื่อ v = 12, 13, 14

การสร้างกลุ่ม L

- จำนวนบิตแมปเวกเตอร์ L ที่ต้องสร้าง = m - 2 = 5-2 = 3 ซึ่งก็คือ (ใช้สัญลักษณ์ L^k ยกเว้นค่าลำดับที่ k = 0) บิตแมปเวกเตอร์ $L^1..L^3$ โดยใช้สูตร

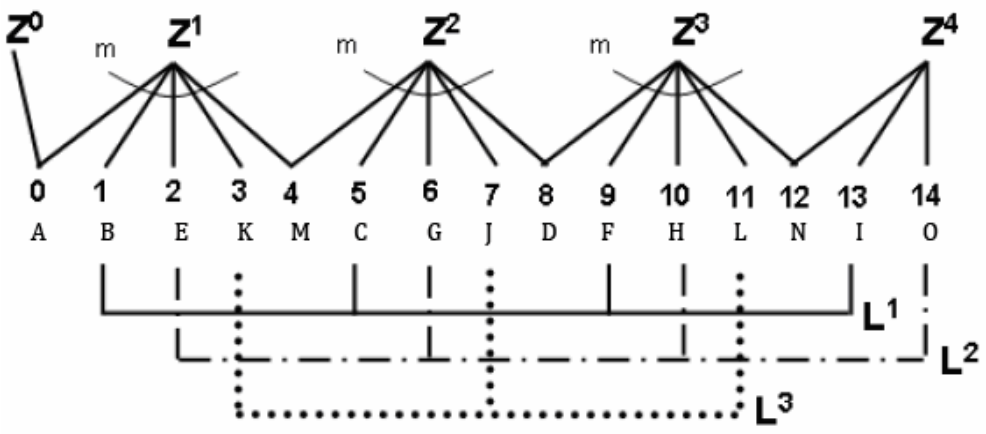
k = v mod (m-1)

- โดยกำหนดให้บิตแมปเวกเตอร์ L^1 แทนค่า 1, 5, 9, 13 เมื่อ v = 1, 5, 9, 13

- บิตแมปเวกเตอร์ L^2 แทนค่า 2, 6, 10, 14 เมื่อ v = 2, 6, 10, 14

- บิตแมปเวกเตอร์ L^3 แทนค่า 3, 7, 11 เมื่อ v = 3, 7, 11

จำนวนบิตแมปเวกเตอร์ทั้งหมดที่ต้องสร้างเท่ากับจำนวนบิตแมปเวกเตอร์ ในกลุ่ม Z + จำนวนบิตแมปเวกเตอร์ในกลุ่ม L จะได้ว่า 5 + 3 = 8 ดังนั้นการลงรหัสของดัชนีบิตแมปแบบกระจายที่ได้จากกระบวนการ SBIOC จะได้ดังภาพประกอบ 4-5



ภาพประกอบ 4-4 รูปแบบทั่วไปของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูลบนแอทริบิวต์ Y (มีค่า C = 15)

RID	แอทริบิวต์ Y
1	10 (H)
2	8 (D)
3	11 (L)
4	5 (C)
5	8 (D)
6	1 (B)
7	12 (N)
8	0 (A)
9	6 (G)
:	:
100	9 (F)

L^3	L^2	L^1	Z^4	Z^3	Z^2	Z^1	Z^0
0	1	0	0	1	0	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	0	0	1	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1
0	1	0	0	0	1	0	0
:	:	:	:	:	:	:	:
0	0	1	0	1	0	0	0

ภาพประกอบ 4-5 รูปแบบการลงรหัสของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูลบนแอทริบิวต์ Y

4.2 การหาค่าเหมาะสมที่สุดของการกำหนดค่าน้ำหนักข้อมูล

จากการศึกษางานวิจัยที่ผ่านมา จะเห็นว่าโครงสร้างของดัชนีบิตแมปแบบกระจาย [3] ยังสามารถเพิ่มความเร็วในการค้นหาข้อมูลได้อีก โดยการพิจารณาการค้นหาข้อมูลที่อยู่ในกลุ่มสมาชิกเดียวกัน โดยได้เสนอเทคนิคการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้น โดยใช้การแบ่งกลุ่มข้อมูล (Data Clustering) [5,6,7,9,10,17] ซึ่งเป็นหนึ่งในเทคนิคการทำเหมืองข้อมูล (Data Mining) [5,6,7,10,11,13,17] มาช่วยในการจัดกลุ่มและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจาย [3,11]

ในส่วนนี้จะเป็นการกล่าวถึงการให้น้ำหนักกับข้อมูลโดยพิจารณาความสำคัญของข้อมูลเป็นหลัก โดยจะมีตัวอย่างการพิจารณาดัชนีบิตแมปแบบกระจายแบบทั่วไปและแบบพิจารณากลุ่มเดียวกัน ด้วยค่า m ค่าต่าง ๆกัน ดังตัวอย่างต่อไปนี้

1) ตาราง 4-2 แสดงตารางพิจารณาการให้น้ำหนัก (กรณี m=3) โดยเปรียบเทียบคู่กันระหว่างการพิจารณาการสอบถามข้อมูลของดัชนีบิตแมปแบบกระจายแบบทั่วไปและดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกัน (กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 1 จะไม่นำมาพิจารณาเนื่องจากถือว่าเป็นการค้นหาแบบเท่ากัน) กล่าวคือ

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 2 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 4 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ดังนั้นการอ่านข้อมูลจะอ่าน 4 ครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 2 item จะได้ว่าจะมีการ AND 2 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆมีค่าเท่ากับ 2 จะมีการอ่านข้อมูล 4 ครั้ง และดำเนินการตรรกะ AND 2 ครั้ง OR 1 ครั้งสำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีค่าเท่ากันดัชนีบิตแมปแบบกระจายแบบทั่วไปเนื่องจากจำนวนสมาชิกไม่เท่ากับค่า m

ตาราง 4-2 ตารางพิจารณาการให้น้ำหนัก (กรณี m=3)

จำนวนสมาชิก item ใน Query ใด ๆ	ดัชนีบิตแมปแบบกระจาย		ดัชนีบิตแมปแบบกระจาย พิจารณากลุ่มเดียวกัน	
	อ่านข้อมูล	ดำเนินการตรรกะ	อ่านข้อมูล	ดำเนินการตรรกะ
1	2	1 AND	2	1 AND
2	4	2 AND, 1 OR	4	2 AND,1 OR
3	6	3 AND, 2 OR	1	0
4	8	4 AND, 3 OR	1+2	1 AND,1 OR
5	10	5 AND, 4 OR	1+4	2 AND,2 OR
6	12	6 AND, 5 OR	2	1 OR

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 3 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 6 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 3 item จะได้ว่าจะมีการ AND 3 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 3 จะมีการอ่านข้อมูล 6 ครั้ง และ ดำเนินการตรรกะ AND 3 ครั้ง OR 2 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 1 ครั้งเท่านั้น และไม่มีการดำเนินการตรรกะใด ๆก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 5 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 10 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 5 item จะได้ว่าจะมีการ AND 5 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 5 จะมีการอ่านข้อมูล 10 ครั้ง และ ดำเนินการตรรกะ AND 5 ครั้ง OR 4 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 5 ครั้ง เนื่องจากจะอ่านข้อมูล 1 ครั้งสำหรับกลุ่มเดียวกันที่มี m = 3 และอีก 2 item ที่เหลือจะอ่านข้อมูล 4 ครั้ง ดำเนินการตรรกะ AND 2 ครั้ง และ OR เพียง 1 ครั้ง หลังจากนั้นนำผลที่ได้มาดำเนินการตรรกะ OR อีกครั้ง ก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 6 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 12 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 6 item จะได้ว่าจะมีการ AND 6 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 6 จะมีการอ่านข้อมูล 12 ครั้ง และ ดำเนินการตรรกะ AND 6 ครั้ง OR 5 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 2 ครั้งและดำเนินการตรรกะ OR เพียง 1 ครั้งก็จะได้คำตอบทันที

2) ตาราง 4-3 แสดงตารางพิจารณาการให้น้ำหนัก (กรณี m=4) โดยเปรียบเทียบคู่กันระหว่างการพิจารณาการสอบถามข้อมูลของดัชนีบิตแมปแบบกระจายแบบทั่วไปและดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกัน (กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 1 จะไม่นำมาพิจารณาเนื่องจากถือว่าเป็นการค้นหาแบบเท่ากัน) กล่าวคือ

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 2 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 4 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ดังนั้นการอ่านข้อมูลจะอ่าน 4 ครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 2 item จะได้ว่าจะมีการ AND 2 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆมีค่าเท่ากับ 2 จะมีการอ่านข้อมูล 4 ครั้ง

ตาราง 4-3 ตารางพิจารณาการให้น้ำหนัก (กรณี m=4)

จำนวนสมาชิก item ใน Query ใด ๆ	ดัชนีบิตแมปแบบกระจาย		ดัชนีบิตแมปแบบกระจาย พิจารณากลุ่มเดียวกัน	
	อ่านข้อมูล	ดำเนินการตรรกะ	อ่านข้อมูล	ดำเนินการตรรกะ
1	2	1 AND	2	1 AND
2	4	2 AND, 1 OR	4	2 AND, 1 OR
3	6	3 AND, 2 OR	6	3 AND, 2 OR
4	8	4 AND, 3 OR	1	0
5	10	5 AND, 4 OR	1+2	1 AND, 1 OR
6	12	6 AND, 5 OR	1+4	2 AND, 2 OR
7	14	7 AND, 6 OR	1+6	3 AND, 3 OR
8	16	8 AND, 7 OR	2	1 OR

ดำเนินการตรรกะ AND 2 ครั้ง OR 1 ครั้งสำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีค่าเท่ากันดัชนีบิตแมปแบบกระจายแบบทั่วไปเนื่องจากจำนวนสมาชิกไม่เท่ากับค่า m

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 4 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 8 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 4 item จะได้ว่าจะมีการ AND 4 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน Query ใด ๆ มีค่าเท่ากับ 4 จะมีการอ่านข้อมูล 8 ครั้ง และ ดำเนินการตรรกะ AND 4 ครั้ง OR 3 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 1 ครั้งเท่านั้นและไม่มีการดำเนินการตรรกะใด ๆก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 7 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 14 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 7 item จะได้ว่าจะมีการ AND 7 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 7 จะมีการอ่านข้อมูล 14 ครั้ง และ ดำเนินการตรรกะ AND 7 ครั้ง OR 6 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 7 ครั้ง เนื่องจากจะอ่านข้อมูล 1 ครั้งสำหรับกลุ่มเดียวกันที่มี m = 4 และอีก 3 item ที่เหลือจะอ่านข้อมูล 6 ครั้ง ดำเนินการตรรกะ AND 3 ครั้ง และ OR เพียง 2 ครั้ง หลังจากนั้นนำผลที่ได้มาดำเนินการตรรกะ OR อีกครั้ง ก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 8 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 16 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 8 item จะได้ว่าจะมีการ AND 8 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 8 จะมีการอ่านข้อมูล 16 ครั้ง และ ดำเนินการตรรกะ AND 8 ครั้ง OR 7 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 2 ครั้งและดำเนินการตรรกะ OR เพียง 1 ครั้งก็จะได้คำตอบทันที

3) ตาราง 4-4 แสดงตารางพิจารณาการให้น้ำหนัก (กรณี m=5) โดยเปรียบเทียบคู่กันระหว่างการพิจารณาการสอบถามข้อมูลของดัชนีบิตแมปแบบกระจายแบบทั่วไปและดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกัน (กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 1 จะไม่นำมาพิจารณาเนื่องจากถือว่าเป็นการค้นหาแบบเท่ากัน) กล่าวคือ

- กรณีจำนวนสมาชิก item ใน query ใด ๆ มีค่าเท่ากับ 2 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 4 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ดังนั้นการอ่านข้อมูลจะอ่าน 4 ครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 2 item จะได้ว่าจะมีการ AND 2 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใด ๆมีค่าเท่ากับ 2 จะมีการอ่านข้อมูล 4 ครั้ง และดำเนินการตรรกะ AND 2 ครั้ง OR 1 ครั้งสำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีค่าเท่ากันดัชนีบิตแมปแบบกระจายแบบทั่วไปเนื่องจากจำนวนสมาชิกไม่เท่ากับค่า m

ตาราง 4-4 ตารางพิจารณาการให้น้ำหนัก (กรณี m=5)

จำนวนสมาชิก item ใน Query ใด ๆ	ดัชนีบิตแมปแบบกระจาย		ดัชนีบิตแมปแบบกระจาย พิจารณากลุ่มเดียวกัน	
	อ่านข้อมูล	ดำเนินการตรรกะ	อ่านข้อมูล	ดำเนินการตรรกะ
1	2	1 AND	2	1 AND
2	4	2 AND, 1 OR	4	2 AND, 1 OR
3	6	3 AND, 2 OR	6	3 AND, 2 OR
4	8	4 AND, 3 OR	8	4 AND, 3 OR
5	10	5 AND, 4 OR	1	0
6	12	6 AND, 5 OR	1+2	1 AND, 1 OR
7	14	7 AND, 6 OR	1+4	2 AND, 2 OR
8	16	8 AND, 7 OR	1+6	3 AND, 3 OR
9	18	9 AND, 8 OR	1+8	4 AND, 4 OR
10	20	10 AND, 9 OR	2	1 OR

- กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 5 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 10 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ 1 ครั้ง ดังนั้น ในที่นี้มี 5 item จะได้ว่าจะมีการ AND 5 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 5 จะมีการอ่านข้อมูล 10 ครั้ง และ ดำเนินการตรรกะ AND 5 ครั้ง OR 4 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 1 ครั้งเท่านั้นและไม่มีการดำเนินการตรรกะใดๆก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 9 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 18 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 9 item จะได้ว่าจะมีการ AND 9 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 9 จะมีการอ่านข้อมูล 18 ครั้ง และ ดำเนินการตรรกะ AND 9 ครั้ง OR 8 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 9 ครั้ง เนื่องจากจะอ่านข้อมูล 1 ครั้งสำหรับกลุ่มเดียวกันที่มี m = 5 และอีก 4 item ที่เหลือจะอ่านข้อมูล 8 ครั้ง และดำเนินการตรรกะ AND 4 ครั้งและ OR เพียง 3 ครั้ง หลังจากนั้นนำผลที่ได้มาดำเนินการตรรกะ OR อีกครั้ง ก็จะได้คำตอบทันที

- กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 10 ดัชนีบิตแมปแบบกระจายจะอ่านข้อมูล 20 ครั้งเพราะแต่ละ item ต้องอ่านค่า Z และ L อย่างละครั้ง ในส่วนของการดำเนินการตรรกะของแต่ละ item จะมีการ AND กัน item ละ1 ครั้ง ดังนั้น ในที่นี้มี 10 item จะได้ว่าจะมีการ AND 10 ครั้งและนำผลที่ได้มา OR กัน ดังนั้นสรุปได้ว่า กรณีจำนวนสมาชิก item ใน query ใดๆ มีค่าเท่ากับ 10 จะมีการอ่านข้อมูล 20 ครั้ง และ ดำเนินการตรรกะ AND 10 ครั้ง OR 9 ครั้ง สำหรับดัชนีบิตแมปแบบกระจายพิจารณากลุ่มเดียวกันจะมีการอ่านข้อมูล 2 ครั้งและดำเนินการตรรกะ OR เพียง 1 ครั้งก็จะได้คำตอบทันที

จากตัวอย่างที่ผ่านมาสรุปได้ว่า จำนวนสมาชิก item ใน query ใดๆถ้ามีจำนวนรวมเท่ากับค่า m หรือ เท่ากับจำนวนเท่าของค่า m จะทำให้ลดจำนวนการอ่านข้อมูลและการดำเนินการตรรกะได้อย่างมาก ดังนั้นงานวิจัยชิ้นนี้จะให้ความสำคัญในกรณีดังกล่าวเป็นพิเศษ ในทำนองเดียวกันจะให้ความสำคัญรองลงมาในกรณีที่ จำนวนสมาชิก item ใน query ใดๆ ถ้ามีจำนวนรวมไม่เท่ากับค่า m หรือไม่เท่ากับจำนวนเท่าของค่า m และ ไม่ให้ความสำคัญกับกรณีที่มีการสอบถามข้อมูลแบบเท่ากัน ดังแสดงไว้ที่ ตาราง 4-5

ตาราง 4-5 การกำหนดค่าน้ำหนักของแต่ละ item

น้ำหนักของ Item	คำอธิบาย
0	สมาชิก item ที่ปรากฏใน query ใดๆ มี 1 item หรือ ไม่ปรากฏใน query
2	สมาชิก item ที่ปรากฏใน query ใดๆ เมื่อรวมกันไม่เท่ากับจำนวนเท่าของ m ไม่เท่ากับ 1 และ ไม่เท่ากับ 0
4	สมาชิก item ที่ปรากฏใน query ใดๆ เมื่อรวมกันเท่ากับจำนวนเท่าของ m (rm เมื่อ r=1, 2, 3 ..)

Q1: SELECT * FROM T WHERE Y in (A, B, C, E, J, K, M, N)

Q2: SELECT * FROM T WHERE Y in (D, F, H, L, N)

Q3: SELECT * FROM T WHERE Y in (G, J)

Q4: SELECT * FROM T WHERE Y in (A, B, D, E, F, H, K, L, M, N)

ภาพประกอบ 4-6 ตัวอย่าง workload

RID	แอทริบิวต์ Y	L^3	L^2	L^1	Z^4	Z^3	Z^2	Z^1	Z^0
1	10 (H)	0	1	0	0	1	0	0	0
2	8 (D)	0	0	0	0	1	1	0	0
3	11 (L)	1	0	0	0	1	0	0	0
4	5 (C)	0	0	1	0	0	1	0	0
5	8 (D)	0	0	0	0	1	1	0	0
6	1 (B)	0	0	1	0	0	0	1	0
7	12 (N)	0	0	0	1	1	0	0	0
8	0 (A)	0	0	0	0	0	0	1	1
9	6 (G)	0	1	0	0	0	1	0	0
:	:	:	:	:	:	:	:	:	:
100	9 (F)	0	0	1	0	1	0	0	0

ภาพประกอบ 4-7 รูปแบบการลงรหัสของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูลบนแอทริบิวต์ Y

4.3 ขั้นตอนวิธีการสอบถามข้อมูลแบบค่าเท่ากัน

การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลมีรูปแบบทั่วไปคือ

$$"Y = v =" \begin{cases} Z^{v/(m-1)} \wedge Z^{v/(m-1)+1} & \text{ถ้า } v \mod(m-1) = 0 \\ Z^{v/(m-1)+1} \wedge L^{v \bmod (m-1)} & \text{กรณีอื่น ๆ} \end{cases}$$

ตัวอย่างเช่น จากภาพประกอบ 4-7 ต้องการทราบว่าบนแอทริบิวต์ Y มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ H กล่าวคือ จากภาพประกอบ 4-7 H มีค่าเท่ากับ 10 ซึ่งตรงกับเงื่อนไขที่ 2 นั้นคือ (10mod(5-1) มีค่าไม่เท่ากับ 0) ดังนั้น คำตอบสามารถหาได้จาก

$$\begin{aligned} "Y = 10" &= Z^{v/(m-1)+1} \wedge L^{v \bmod (m-1)} \\ &= Z^{10/(5-1)+1} \wedge L^{10 \bmod (5-1)} \\ &= Z^3 \wedge L^2 \end{aligned}$$

ซึ่งแบ่งเป็นขั้นตอนเพื่อหาคำตอบได้ดังนี้

1) อ่านบิตแมปเวกเตอร์ Z^3

2) อ่านบิตแมปเวกเตอร์ L^2

3) ดำเนินการตรรกะ AND บิตต่อบิตระหว่างบิตแมปเวกเตอร์ที่ได้จากข้อ 1) และ 2)

4) คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้จากข้อ 3 พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น 10 คือเรคอร์ดที่ 1 ดังภาพประกอบ 4-8

RID	แอทริบิวต์ Y	L^2		Z^3		$Z^3 \wedge L^2$
1	10 (H)	1		1		1
2	8 (D)	0		1		0
3	11 (L)	0		1		0
4	5 (C)	0		0		0
5	8 (D)	0	AND	1	=	0
6	1 (B)	0		0		0
7	12 (N)	0		1		0
8	0 (A)	0		0		0
9	6 (G)	1		0		0
:	:	:		:		:
100	9 (F)	0		1		0

ภาพประกอบ 4-8 ขั้นตอนวิธีการสอบถามข้อมูลแบบค่าเท่ากันของ SBIOC บนแอทริบิวต์ Y (กรณีสอบถามข้อมูล H)

4.4 ขั้นตอนวิธีการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน

ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันบนดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีรูปแบบทั่วไปเหมือนกับการสอบถามแบบค่าเท่ากัน โดยที่สามารถทำได้โดยการหาคำตอบของแต่ละสมาชิก แล้วนำผลที่ได้มาดำเนินการตรรกะ OR

ตัวอย่างเช่น จากภาพประกอบ 4-6 ถ้าต้องการค้นหาของ Q3 นั้นคือ Y = G และ J กล่าวคือ จากภาพประกอบ 4-7 จะได้ว่า Y = 6 และ 7

<u>วิธีการดึงข้อมูล</u>

1) ตรวจสอบดูว่า Y = 6 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{6}{5-1} \right\rfloor + 1 = 2$

2) ตรวจสอบดูว่า Y = 6 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 6 mod (5-1) = 2

3) จะได้ว่า Y = 6 อยู่ในกลุ่ม Z^2 และ L^2 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^2 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^2$ ที่มีค่าบิตเท่ากับ 1

4) ตรวจสอบดูว่า Y = 7 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{7}{5-1} \right\rfloor + 1 = 2$

5) ตรวจสอบดูว่า Y = 7 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m−1) = 7 mod (5−1) = 3

6) จะได้ว่า Y = 7 อยู่ในกลุ่ม Z^2 และ L^3 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^3 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^3$ ที่มีค่าบิตเท่ากับ 1

7) นำผลที่ได้มา OR กัน จะได้ว่า $\left(Z^2 \wedge L^2\right) \vee \left(Z^2 \wedge L^3\right)$ คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้ พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น G และ J คือเรคอร์ดที่ 9 ดังภาพประกอบ 4-9

RID	แอทริบิวต์ Y	Z^2		L^2		Z^2		L^3		$\left(Z^2 \wedge L^2\right) \vee \left(Z^2 \wedge L^3\right)$
1	10 (H)	0		1		0		0		0
2	8 (D)	1		0		1		0		0
3	11 (L)	0		0		0		1		0
4	5 (C)	1		0		1		0		0
5	8 (D)	(1	AND	0	)OR(	1	AND	0	)	0
6	1 (B)	0		0		0		0		0
7	12 (N)	0		0		0		0		0
8	0 (A)	0		0		0		0		0
9	6 (G)	1		1		1		0		1
:	:	:		:		:		:		:
100	9 (F)	0		0		0		0		0

ภาพประกอบ 4-9 ขั้นตอนการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน (Y = G และ J)

4.5 ขั้นตอนวิธีการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน

สำหรับการค้นหาข้อมูลแบบความเป็นสมาชิกที่อยู่กลุ่มเดียวกัน สามารถทำได้โดยการค้นหาทีละค่า แล้วมาดำเนินการตรรกะ OR ถ้าข้อมูลนั้นเป็นสมาชิกอยู่ภายในกลุ่ม Z เดียวกัน ก็สามารถอ่านเฉพาะบิตแมปเวกเตอร์ $Z^{v/(m-1)+1}$ เพียงบิตแมปเวกเตอร์เดียว

- **ตัวอย่างที่ 1** จากภาพประกอบ 4-6 ถ้าต้องการค้นหาของ Q1 นั้นคือ Y = A, B, C, E, J, K, M และ N กล่าวคือ จากภาพประกอบ 4-7 จะได้ว่า Y = 0, 1, 5, 2, 7, 3, 4 และ 12

<u>วิธีการดึงข้อมูล</u>

1) การค้นหาของ Q1 นี้ สามารถแยกได้เป็น 2 ชุด กล่าวคือ ชุดที่ 1 มี 0, 1, 2, 3 และ 4 เนื่องจากอยู่ในกลุ่มเดียวกัน ชุดที่ 2 มี 5, 7 และ 12

2) ตรวจสอบดูว่า Y = 0, 1, 2, 3 และ 4 อยู่ในกลุ่ม Z ใด ซึ่งจะได้ว่าอยู่ที่กลุ่มของ Z^1

3) ตรวจสอบดูว่า Y = 5 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{5}{5-1} \right\rfloor + 1 = 2$

4) ตรวจสอบดูว่า Y = 5 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 5 mod (5-1) = 1

5) จะได้ว่า Y = 5 อยู่ในกลุ่ม Z^2 และ L^1 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^1 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^1$ ที่มีค่าบิตเท่ากับ 1

6) ตรวจสอบดูว่า Y = 7 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{7}{5-1} \right\rfloor + 1 = 2$

7) ตรวจสอบดูว่า Y = 7 อยู่ในกลุ่ม L^k ใด โดย k = v mod (m-1) = 7 mod (5-1) = 3

8) จะได้ว่า Y = 7 อยู่ในกลุ่ม Z^2 และ L^3 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^2 ดำเนินการตรรกะ AND กับ L^3 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^2 \wedge L^3$ ที่มีค่าบิตเท่ากับ 1

9) ตรวจสอบดูว่า Y = 12 อยู่ในกลุ่ม Z^j ใด โดย j = $\left\lfloor \frac{v}{m-1} \right\rfloor + 1$ = $\left\lfloor \frac{12}{5-1} \right\rfloor + 1 = 4$ และพบอีกว่า หารด้วย m-1 ลงตัว อยู่ในกลุ่ม Z ทั้ง 2 กลุ่ม คือ กลุ่ม Z^{j-1} และ Z^j ดังนั้น

10) ตรวจสอบดูว่า $Z^{j-1} = Z^{4-1}$ จะได้เป็น Z^3

11) จะได้ว่า Y = 12 อยู่ในกลุ่ม Z^3 และ Z^4 ดังนั้นจึงอ่านบิตแมปเวกเตอร์ Z^3 ดำเนินการตรรกะ AND กับ Z^4 ผลลัพธ์ที่ได้ คือ ผลลัพธ์ของ $Z^3 \wedge Z^4$ ที่มีค่าบิตเท่ากับ 1

12) นำผลที่ได้มา OR กัน จะได้ว่า $Z^1 \vee (Z^2 \wedge L^1) \vee (Z^2 \wedge L^3) \vee (Z^3 \wedge Z^4)$ คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้ พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น A, B, C, E, J, K, M และ N คือเรคอร์ดที่ 4, 6, 7 และ 8 ดังภาพประกอบ 4-10

RID	แอทริบิวต์ Y	Z^1		Z^2		L^1		Z^2		L^3		Z^3		Z^4		ผลลัพธ์
1	10 (H)	0		0		0		0		0		1		0		0
2	8 (D)	0		1		0		1		0		1		0		0
3	11 (L)	0		0		0		0		1		1		0		0
4	5 (C)	0		1		1		1		0		0		0		1
5	8 (D)	0	OR(	1	AND	0	)OR(	1	AND	0	)OR(	1	AND	0	⇒	0
6	1 (B)	1		0		1		0		0		0		0		1
7	12 (N)	0		0		0		0		0		1		1		1
8	0 (A)	1		0		0		0		0		0		0		1
9	6 (G)	0		1		0		1		0		0		0		0
:	:	:		:		:		:		:		:		:		0
100	9 (F)	0		0		1		0		0		1		0		0

ภาพประกอบ 4-10 ขั้นตอนการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน (Y = A, B, C, E, J, K, M และ N)

- **ตัวอย่างที่ 2** จากภาพประกอบ 4-6 ถ้าต้องการค้นหาของ Q2 นั้นคือ Y = D, F, H, L และ N กล่าวคือ จากภาพประกอบ 4-7 จำได้ว่า Y = 8, 9, 10, 11 และ 12

<u>วิธีการดึงข้อมูล</u>

การค้นหาของ Q1 นี้ จะสังเกตุว่าเป็นการค้นหาที่อยู่ในกลุ่มเดียวกัน กล่าวคือ 8, 9, 10, 11 และ 12 จะอยู่ที่ Z^3 ดังนั้นสามารถดึงข้อมูล Z^3 ออกมาเป็นคำตอบของการสอบถามได้ทันที ดังภาพประกอบ 4-11

RID	แอทริบิวต์ Y
1	10 (H)
2	8 (D)
3	11 (L)
4	5 (C)
5	8 (D)
6	1 (B)
7	12 (N)
8	0 (A)
9	6 (G)
:	:
100	9 (F)

Z^3
1
1
1
0
1
0
1
0
0
:
1

$\Rightarrow$

ผลลัพธ์
1
1
1
0
1
0
1
0
0
:
1

ภาพประกอบ 4-11 ขั้นตอนการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน (Y = D, F, H, L และ N)

- **ตัวอย่างที่ 3** จากภาพประกอบ 4-6 ถ้าต้องการค้นหาของ Q4 นั้นคือ Y = A, B, D, E, F, H, K, L, M และ N กล่าวคือ จากภาพประกอบ 4-7 จำได้ว่า Y = 0, 1, 8, 2, 9, 10, 3, 11, 4 และ 12

<u>วิธีการดึงข้อมูล</u>

1) การค้นหาของ Q4 นี้ สามารถแยกได้เป็น 2 ชุด กล่าวคือ ชุดที่ 1 มี 0, 1, 2, 3 และ 4 เนื่องจากอยู่ในกลุ่มเดียวกัน ชุดที่ 2 มี 8, 9, 10, 11 และ 12

2) การค้นหาของข้อมูลชุดที่ 1 นี้ จะสังเกตุว่าเป็นการค้นหาที่อยู่ในกลุ่มเดียวกัน กล่าวคือ 0, 1, 2, 3 และ 4 จะอยู่ที่ Z^1 ดังนั้นสามารถดึงข้อมูล Z^1 ออกมาเป็นคำตอบของการสอบถามได้ทันที

3) การค้นหาของข้อมูลชุดที่ 2 นี้ จะสังเกตุว่าเป็นการค้นหาที่อยู่ในกลุ่มเดียวกัน กล่าวคือ 8, 9, 10, 11 และ 12 จะอยู่ที่ Z^3 ดังนั้นสามารถดึงข้อมูล Z^3 ออกมาเป็นคำตอบของการสอบถามได้ทันที

4) นำผลที่ได้มา OR กัน จะได้ว่า $\left(Z^1 \vee Z^3\right)$ คำตอบที่ได้พิจารณาจากบิตแมปเวกเตอร์ผลลัพธ์ที่ได้ พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น A, B, D, E, F, H, K, L, M และ N คือเรคอร์ดที่ 1, 2, 3, 5, 6, 7, 8 และ 100 ดังภาพประกอบ 4-12

RID	แอทริบิวต์ Y	Z^1		Z^3		ผลลัพธ์
1	10 (H)	0		1		1
2	8 (D)	0		1		1
3	11 (L)	0		1		1
4	5 (C)	0		0		0
5	8 (D)	0	OR	1	⇒	1
6	1 (B)	1		0		1
7	12 (N)	0		1		1
8	0 (A)	1		0		1
9	6 (G)	0		0		0
:	:	:		:		0
100	9 (F)	0		1		1

ภาพประกอบ 4-12 ขั้นตอนการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน (Y = A, B, D, E, F, H, K, L, M และ N)

4.6 ข้อดีของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ประหยัดพื้นที่ได้มากกว่าดัชนีบิตแมปแบบพื้นฐาน และดัชนีบิตแมปแบบช่วง

นอกจากนี้ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะค้นหาข้อมูลได้เร็วกว่าดัชนีบิตแมปแบบช่วง และ แบบเข้ารหัส อย่างไรก็ตาม ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลจะใช้เวลาน้อยกว่าดัชนีบิตแมปแบบกระจายทั่วไป กรณีในการค้นหาข้อมูลแบบความเป็นสมาชิกที่อยู่ในกลุ่มเดียวกัน โดยที่สมาชิก item ที่ปรากฏใน query ใด ๆ เมื่อรวมกันเท่ากับจำนวน m หรือ เท่ากับจำนวนเท่าของ m (rm เมื่อ r=1, 2, 3 ..) หรือค่าที่ต้องการค้นหานั้นอยู่ในกลุ่ม Z หรือ L เดียวกัน เพราะบิตแมปเวกเตอร์ที่ถูกอ่านมีจำนวน 1 บิตแมปเวกเตอร์เท่านั้น

4.7 ข้อจำกัดของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้พื้นที่ในการจัดเก็บดัชนีมากกว่าดัชนีบิตแมปแบบเข้ารหัส และ พื้นที่ในการจัดเก็บดัชนีเท่ากับดัชนีบิตแมปแบบกระจายทั่วไป เนื่องจากโครงสร้างในการจัดเก็บดัชนียังเป็นเหมือนเดิมกับต้นแบบ นั้นคือ ดัชนีบิตแมปแบบกระจายทั่วไป ในทำนองเดียวกัน กรณีที่มีการค้นหาข้อมูลแบบเท่ากัน หรือ

แบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลยังจะใช้เวลาในการค้นหาข้อมูลเท่ากันกับดัชนีบิตแมปแบบกระจายทั่วไปอยู่

กรณีที่ workload มาไม่ถูกต้อง (GIGO : Garbage In, Garbage Out) อาจทำให้การแบ่งกลุ่มข้อมูลผิด ส่งผลให้การทำดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลผิดพลาดไปด้วย แต่อย่างไรก็ตามประสิทธิภาพในการค้นหาข้อมูลก็จะเท่ากับดัชนีบิตแมปแบบกระจายทั่วไปอยู่

ในภาพรวม ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลจะมีประสิทธิภาพและเหมาะกับการสอบถามข้อมูลแบบแบบความเป็นสมาชิกที่อยู่ในกลุ่มเดียวกันมากที่สุด เมื่อเปรียบเทียบกับดัชนีบิตแมปแบบพื้นฐาน ดัชนีบิตแมปแบบช่วง ดัชนีบิตแมปแบบเข้ารหัส และ ดัชนีบิตแมปแบบกระจายทั่วไป

บทที่ 5

การวิเคราะห์และผลการทดลอง

สำหรับบทนี้จะกล่าวถึง การวิเคราะห์เปรียบเทียบค่าใช้จ่ายเกี่ยวกับพื้นที่ที่ใช้ในการจัดเก็บดัชนี ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน และค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล

เป็นที่ทราบกันดีอยู่แล้วว่า ในการสร้างดัชนีแบบบิตแมปขึ้นมาเพื่อรองรับการสอบถามข้อมูลนั้น ปัจจัยหนึ่งที่เราจะต้องคำนึงถึงก็คือค่าใช้จ่าย โดยในงานวิทยานิพนธ์นี้ค่าใช้จ่ายที่เกิดขึ้นในการสร้างดัชนีบิตแมปแต่ละแบบนั้น สามารถพิจารณาได้ 2 วิธีด้วยกัน คือ จากการวิเคราะห์ และจากผลการทดลอง โดยค่าใช้จ่ายที่ได้จากแต่ละวิธีนั้น จะพิจารณาในประเด็นของพื้นที่ที่ใช้ในการจัดเก็บดัชนี และเวลาที่ใช้ในการค้นหาข้อมูล

5.1 ค่าใช้จ่ายจากการวิเคราะห์ (Analytical Method)

ค่าใช้จ่ายที่ใช้ในการสร้างดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลนั้น สามารถวิเคราะห์ได้เป็น 2 ประเด็น คือ พื้นที่ที่ใช้ในการจัดเก็บดัชนีและเวลาที่ใช้ในการค้นหาข้อมูล

5.1.1 พื้นที่ที่ใช้ในการจัดเก็บดัชนี

กำหนดให้ ตารางเชิงความสัมพันธ์มีจำนวนเรคอร์ดทั้งหมดเท่ากับ N เรคอร์ด แอทริบิวต์ Y ที่จะนำมาสร้างดัชนีมีคาร์ดินอลิตี้เท่ากับ C ดังนี้

$T = \{t_0, t_1, .., t_{N-1}\}$

$Y = \{y_0, y_1, .., y_{C-1}\}$

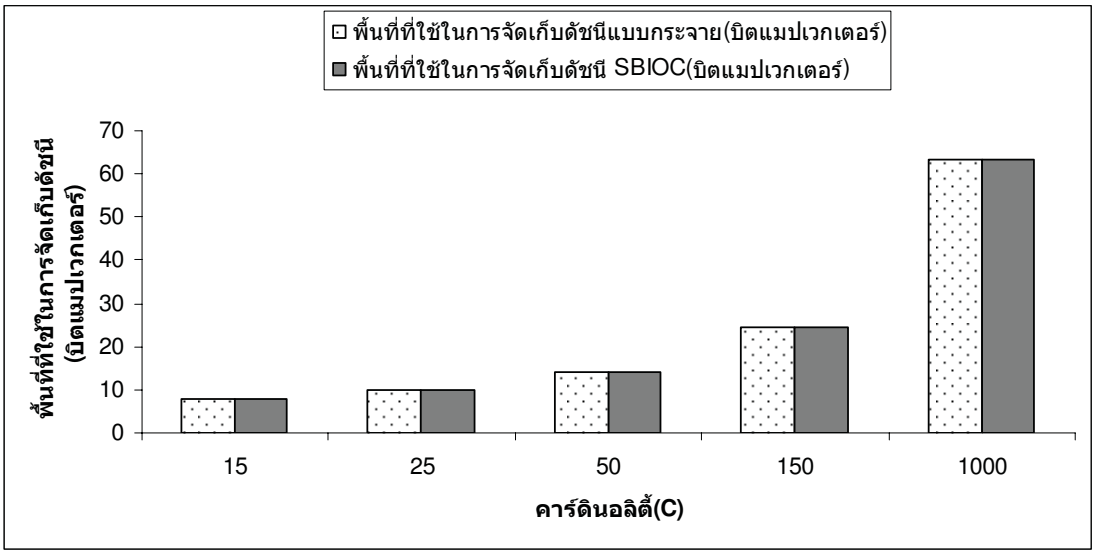
ดังนั้นจะได้ว่า ดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (SBIOC) จะใช้พื้นที่ในการจัดเก็บดัชนี ดังตาราง 5-1

ตาราง 5-1 พื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปทั้ง 2 แบบ เมื่อ C คือ คาร์ดินอลิตี้ของแอทริบิวต์ และ N คือ จำนวนเรคอร์ดของแอทริบิวต์

ชนิดของดัชนีบิตแมป	พื้นที่ที่ใช้ในการจัดเก็บดัชนี
ดัชนีบิตแมปแบบกระจายทั่วไป	$\lceil 2\sqrt{C} \rceil N$
ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	$\lceil 2\sqrt{C} \rceil N$

จากตาราง 5-1 จะเห็นได้ว่า พื้นที่ที่ใช้ในการจัดเก็บดัชนีจะแปรผันโดยตรงกับค่าคาร์ดินอลิตี้ของแอทริบิวต์ที่นำมาสร้างดัชนี เมื่อเปรียบเทียบพื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปทั้ง 2 แบบ จะเห็นได้ว่า มีการใช้พื้นที่เท่ากัน ดังนั้นเพื่อให้เห็นภาพชัดเจนยิ่งขึ้นในการเปรียบเทียบให้เห็นถึงพื้นที่ที่ใช้ในการจัดเก็บดัชนีทั้ง 2 แบบ จึงขอยกตัวอย่างดังภาพประกอบ 5-1

จากภาพประกอบ 5-1 เป็นกราฟที่สร้างขึ้นเพื่อให้เห็นว่า พื้นที่ที่ใช้ในการจัดเก็บดัชนีจะแปรผันโดยตรงกับค่าคาร์ดินอลิตี้ของแอทริบิวต์ที่นำมาสร้างดัชนี ซึ่งค่าคาร์ดินอลิตี้อยู่ในช่วง 15 ถึง 1,000 เมื่อเปรียบเทียบพื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปทั้ง 2 แบบ จะเห็นได้ว่า มีการใช้พื้นที่ในการจัดเก็บดัชนีเท่ากัน ตัวอย่างเช่น แอทริบิวต์ Y ที่มีค่า C = 1,000 จะใช้พื้นที่ในการจัดเก็บของดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีค่าเท่ากับ 63.24 และ 63.24 บิตแมปเวกเตอร์ตามลำดับ ซึ่งเป็นไปตามความคาดหมายกล่าวคือ ดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้พื้นที่ในการจัดเก็บที่เท่ากัน



ภาพประกอบ 5-1 แผนภาพการเปรียบเทียบการประเมินประสิทธิภาพในเรื่องพื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปทั้ง 2 แบบเมื่อ คาร์ดินอลิตี้ (C) ต่าง ๆกัน

5.1.2 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถาม

วิธีการหนึ่งในการวิเคราะห์เวลาที่ใช้ในการค้นหาข้อมูล คือ พิจารณาจากจำนวนบิตแมปที่ถูกอ่าน (Bitmap Scan) และ จำนวนตัวดำเนินการตรรกะกับบิตแมป (Bitmap Operator) การวิเคราะห์เปรียบเทียบค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามสามารถแยกได้เป็น 3 แบบ ดังนี้

5.1.2.1 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน

ดัชนีบิตแมปทั้งสองชนิดมีรูปแบบการลงรหัสที่เหมือนกัน นอกจากจะทำให้ใช้พื้นที่ในการจัดเก็บดัชนีเท่ากันแล้ว ยังทำให้ขั้นตอนการสอบถามแบบค่าเท่ากันใช้เวลาเท่ากันอีกด้วย ทั้งจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ พิจารณาจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด ดังตาราง 5-2

ตาราง 5-2 จำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด

ชนิดของดัชนีบิตแมป	จำนวนบิตแมปเวกเตอร์ที่อ่าน	จำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์
ดัชนีบิตแมปแบบกระจายทั่วไป	2	1 (1AND)
ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	2	1 (1AND)

จากตาราง 5-2 จะเห็นว่า ในการสอบถามแบบค่าเท่ากัน ดัชนีบิตแมปแบบกระจายทั่วไป มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 2 บิตแมปเวกเตอร์ และ มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ 1 ครั้ง ในทำนองเดียวกัน ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 2 บิตแมปเวกเตอร์ และ มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ 1 ครั้ง

จึงเป็นผลให้การสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด ใช้เวลาในการสอบถามข้อมูลเท่ากัน

5.1.2.2 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน

นอกจากจะพิจารณาค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันแล้ว ในส่วนนี้จะพิจารณาค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน ทั้งจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ การพิจารณาจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด (กรณีตัวอย่าง ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิก ซึ่งมีจำนวนสมาชิก 2 ค่า และไม่ได้อยู่กลุ่มเดียวกัน) ดังตาราง 5-3

ตาราง 5-3 จำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด กรณีตัวอย่างสอบถามแบบสมาชิก 2 ค่า

ชนิดของดัชนีบิตแมป	จำนวนบิตแมปเวกเตอร์ที่อ่าน	จำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์
ดัชนีบิตแมปแบบกระจายทั่วไป	4	3 (2AND, 1OR)
ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	4	3 (2AND, 1OR)

จากตาราง 5-3 จะเห็นว่า ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปแบบกระจายทั่วไป มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 4 บิตแมปเวกเตอร์ และ มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ 3 ครั้ง โดยดำเนินการตรรกะ AND จำนวน 2 ครั้ง และ ดำเนินการตรรกะ OR จำนวน 1 ครั้ง ดังนั้นการดำเนินการตรรกะรวมกันทั้งสิ้นมี 3 ครั้ง ในทำนองเดียวกัน ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 4 บิตแมปเวกเตอร์ และ มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ 3 ครั้ง โดยดำเนินการตรรกะ AND จำนวน 2 ครั้ง และ ดำเนินการตรรกะ OR จำนวน 1 ครั้ง ดังนั้นการดำเนินการตรรกะรวมกันทั้งสิ้นมี 3 ครั้ง

จึงเป็นผลให้การสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด ใช้เวลาในการสอบถามข้อมูลเท่ากัน

5.1.2.3 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน

จากการวิเคราะห์ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน และค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน พบว่ามีค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามเท่ากัน ในส่วนนี้จะกล่าวถึง ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน ทั้งจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ พิจารณาจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด (กรณีตัวอย่าง ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิก ซึ่งมีจำนวนสมาชิก 5 ค่า และอยู่กลุ่มเดียวกัน หรือ มีค่าสมาชิกเท่ากับจำนวน m) ดังตาราง 5-4

ตาราง 5-4 จำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด กรณีตัวอย่างสอบถามแบบสมาชิก 5 ค่า ซึ่งมีค่าสมาชิกเท่ากับจำนวน m

ชนิดของดัชนีบิตแมป	จำนวนบิตแมปเวกเตอร์ที่อ่าน	จำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์
ดัชนีบิตแมปแบบกระจายทั่วไป	10	9 (5AND, 4OR)
ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	1	0

จากตาราง 5-4 จะเห็นว่า ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน ของดัชนีบิตแมปแบบกระจายทั่วไป มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 10 บิตแมปเวกเตอร์ และ มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ 9 ครั้ง โดยดำเนินการตรรกะ AND จำนวน 5 ครั้ง และ ดำเนินการตรรกะ OR จำนวน 4 ครั้ง ดังนั้นการดำเนินการตรรกะรวมกันทั้งสิ้นมี 9 ครั้ง ในทำนองเดียวกัน ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีจำนวนบิตแมปเวกเตอร์ที่อ่าน 1 บิตแมปเวกเตอร์ และ ไม่มีการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์ใด ๆ

จะเห็นได้ว่าการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด ใช้เวลาในการสอบถามข้อมูลต่างกัน ซึ่งเป็นไปตามความคาดหมาย กล่าวคือ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะมีค่าใช้จ่ายในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน น้อยกว่า ดัชนีบิตแมปแบบกระจายทั่วไป

5.2 ค่าใช้จ่ายจากผลการทดลอง (Experimental Result)

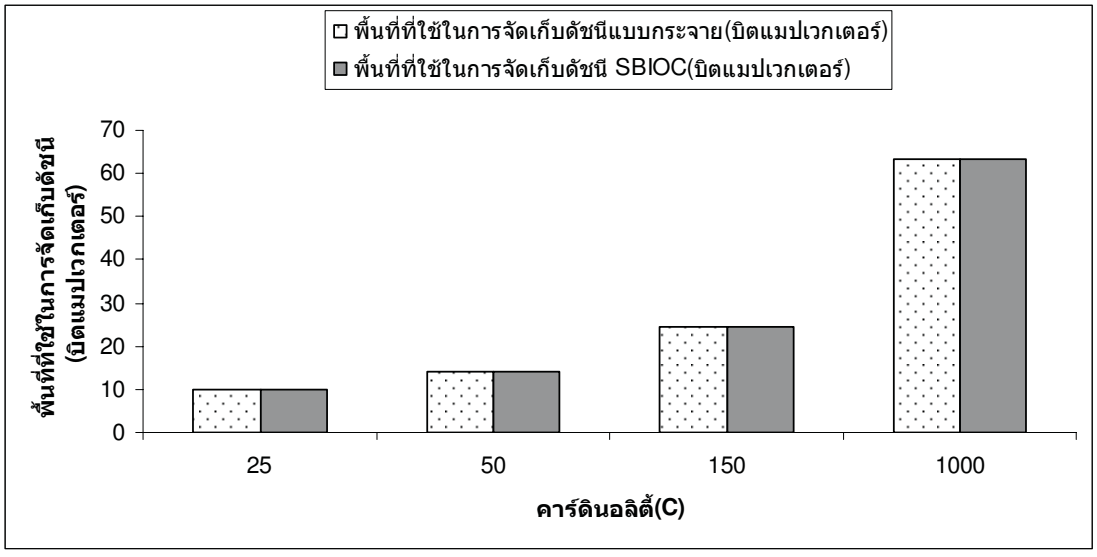
นอกจากเราจะสามารถเปรียบเทียบการประเมินประสิทธิภาพ ของการสร้างดัชนีบิตแมปทั้ง 2 ชนิด ในเรื่องพื้นที่ที่ใช้ในการจัดเก็บดัชนีและเวลาที่ใช้ในการค้นหาข้อมูลแบบต่าง ๆ โดยวิธีการวิเคราะห์ค่าใช้จ่ายแล้วนั้น เรายังสามารถที่จะเปรียบเทียบการประเมินประสิทธิภาพของการสร้างดัชนีและเวลาที่ใช้ในการค้นหาข้อมูลได้อีกวิธีหนึ่ง ซึ่งก็คือ จากผลการทดลองสำหรับงานวิทยานิพนธ์ชิ้นนี้ ได้ทำการทดลองกับดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล โดยทำการประเมินประสิทธิภาพของการสร้างดัชนีในเรื่องพื้นที่ที่ใช้ในการจัดเก็บดัชนีและเวลาที่ใช้ในการค้นหาข้อมูล ต่อไปเราจะทำการเปรียบเทียบค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันจากผลการทดลอง โดยทำการทดลองบนเครื่องคอมพิวเตอร์รุ่น Intel(R) Celeron(R) ที่มีหน่วยประมวลผลกลางขนาด 1.69 GHz

หน่วยความจำ 512 MB ระบบปฏิบัติการ Fedora Core 6 และใช้ภาษาซีในการเขียนโปรแกรม โดยมีผลการทดลองดังนี้

5.2.1 พื้นที่ที่ใช้ในการจัดเก็บดัชนี

สำหรับงานวิทยานิพนธ์ชิ้นนี้ ได้ทำการทดลองประเมินประสิทธิภาพของดัชนีบิตแมป 2 แบบด้วยกัน คือ ดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ซึ่งจากการนำข้อมูลทดสอบจากการวัดเปรียบเทียบสมรรถนะของ TPC-H [18] ตาราง part แอทริบิวต์ p_brand (ค่า C = 25) จำนวน 200,000 เรคอร์ด, แอทริบิวต์ p_size (ค่า C = 50) จำนวน 200,000 เรคอร์ด, แอทริบิวต์ p_type (ค่า C = 150) จำนวน 200,000 เรคอร์ด และ order แอทริบิวต์ o_clerk (ค่า C = 1,000) จำนวน 1,500,000 เรคอร์ด มาทดลอง สามารถแสดงพื้นที่ที่ใช้ในการจัดเก็บดัชนี ได้ดังภาพประกอบ 5-2

พิจารณาภาพประกอบ 5-2 แสดงความสัมพันธ์ระหว่างพื้นที่ที่ใช้ในการสร้างดัชนีกับคาร์ดินอลิตี้ของแอทริบิวต์ที่เลือกมาทำดัชนีบิตแมปทั้ง 2 ชนิด ซึ่งคาร์ดินอลิตี้มีค่าตั้งแต่ 15 ถึง 1,000 จะเห็นได้ว่าทั้งดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้พื้นที่จัดเก็บดัชนีเท่ากันไม่ว่าจะมีค่าคาร์ดินอลิตี้เท่าใดก็ตาม



ภาพประกอบ 5-2 กราฟแสดงการเปรียบเทียบพื้นที่ที่ใช้ในการสร้างดัชนีบิตแมปทั้ง 2 ชนิด เมื่อแอทริบิวต์ที่นำมาทำดัชนีมี 200,000 เรคอร์ด และ 1,500,000 เรคอร์ด (ในกรณีคาร์ดินอลิตี้เท่ากับ 1,000)

โดยสรุปแล้ว ดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้พื้นที่ในการจัดเก็บที่เท่ากัน

5.2.2 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถาม

การพิจารณาค่าใช้จ่ายในการสร้างดัชนีบิตแมปแบบต่าง ๆ นั้น นอกจากจะพิจารณาในเรื่องพื้นที่ที่ใช้ในการจัดเก็บดัชนีแล้วนั้น ปัจจัยหนึ่งซึ่งสำคัญที่จะต้องพิจารณาด้วย ก็คือ เวลาที่ใช้ในการค้นหาข้อมูล สำหรับงานวิทยานิพนธ์ชิ้นนี้ได้ทำการทดลองเพื่อประเมินประสิทธิภาพของดัชนีบิตแมปทั้ง 2 แบบ คือ ดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ในเรื่องของเวลาที่ใช้ในการค้นหาข้อมูล โดยแบ่งเป็น 3 กรณี คือ ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน และ ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน โดยมีผลการทดลองดังนี้

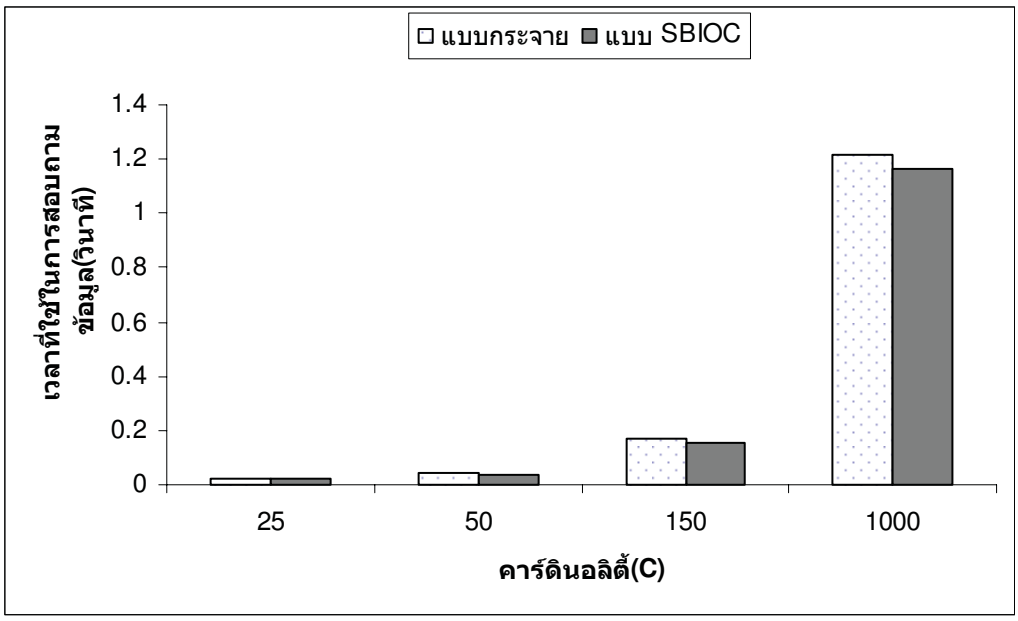
5.2.2.1 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน

งานวิทยานิพนธ์นี้ ได้ทำการทดลองค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน ดำเนินการโดยการรันโปรแกรมเพื่อสอบถามข้อมูล แล้วบันทึกเวลาที่ใช้ ซึ่งดำเนินการทั้งหมด 4 ครั้ง จากนั้นหาค่าเฉลี่ยสำหรับการสอบถามข้อมูล ดังผลการทดลองในตาราง 5-5

จากตาราง 5-5 นำค่าเฉลี่ยของเวลาที่ใช้ในการสอบถามมาเขียนกราฟได้ดังภาพประกอบ 5-3

ตาราง 5-5 เวลาที่ใช้ในการสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปทั้ง 2 ชนิด เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 25, 50, 150 และ 1,000

คาร์ดินอลิตี้ (C) ครั้งที่	เวลาที่ใช้ในการสอบถาม (วินาที)		ความถูกต้องของคำตอบ
	ดัชนีบิตแมปแบบกระจายทั่วไป	ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	
C=25			
ครั้งที่ 1	0.02	0.02	✓
ครั้งที่ 2	0.03	0.02	✓
ครั้งที่ 3	0.02	0.02	✓
ครั้งที่ 4	0.02	0.02	✓
เฉลี่ย	0.0225	0.02	
C=50			
ครั้งที่ 1	0.05	0.04	✓
ครั้งที่ 2	0.04	0.04	✓
ครั้งที่ 3	0.04	0.04	✓
ครั้งที่ 4	0.04	0.04	✓
เฉลี่ย	0.0425	0.04	
C=150			
ครั้งที่ 1	0.17	0.16	✓
ครั้งที่ 2	0.17	0.15	✓
ครั้งที่ 3	0.17	0.15	✓
ครั้งที่ 4	0.18	0.15	✓
เฉลี่ย	0.1725	0.1525	
C=1,000			
ครั้งที่ 1	1.06	0.88	✓
ครั้งที่ 2	0.87	0.85	✓
ครั้งที่ 3	1.46	1.46	✓
ครั้งที่ 4	1.48	1.47	✓
เฉลี่ย	1.2175	1.165	



ภาพประกอบ 5-3 กราฟแสดงการเปรียบเทียบเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด

จากผลการทดลองที่แสดงดังภาพประกอบ 5-3 จะเห็นได้ว่าดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลใช้เวลาในการค้นหาข้อมูลใกล้เคียงกัน ซึ่งสอดคล้องกับ ตาราง 5-2 ซึ่งแสดงจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด

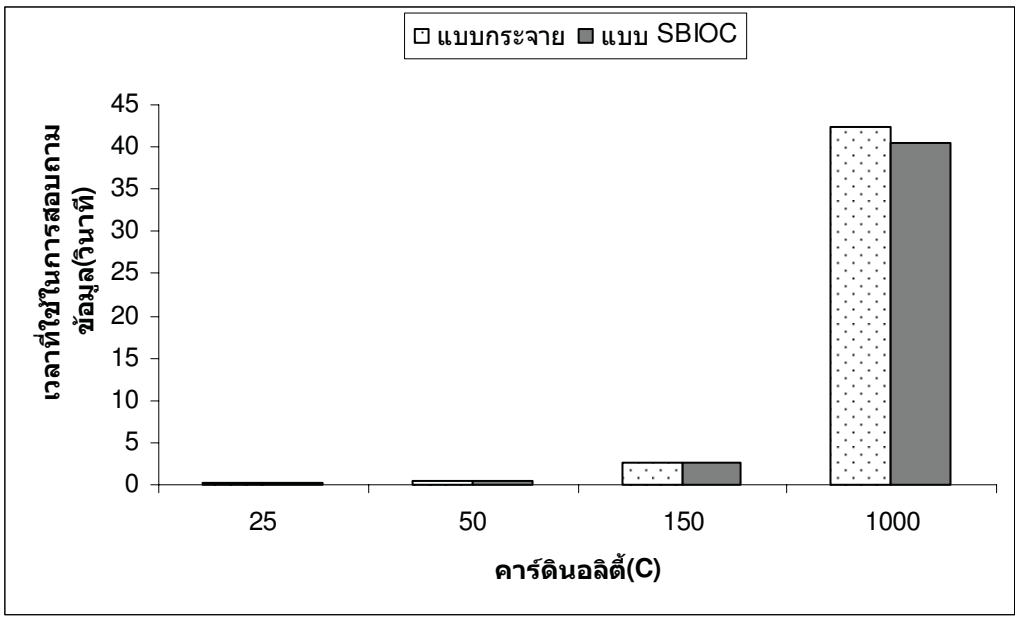
5.2.2.2 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน

การทดลองค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน ดำเนินการโดยการรันโปรแกรมเพื่อสอบถามข้อมูลแล้วบันทึกเวลาที่ใช้ ซึ่งดำเนินการทั้งหมด 4 ครั้ง จากนั้นหาค่าเฉลี่ยสำหรับการสอบถามข้อมูล ดังผลการทดลองในตาราง 5-6

จากตาราง 5-6 นำค่าเฉลี่ยของเวลาที่ใช้ในการสอบถามมาเขียนกราฟได้ดังภาพประกอบ 5-4

ตาราง 5-6 เวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันบนดัชนีบิตแมปทั้ง 2 ชนิด เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 25, 50, 150 และ 1,000 กรณีตัวอย่าง การสอบถามโดยมีค่าสมาชิกเท่ากับจำนวน m-1

คาร์ดินอลิตี้ (C) ครั้งที่	เวลาที่ใช้ในการสอบถาม (วินาที)		ความถูกต้องของคำตอบ
	ดัชนีบิตแมปแบบกระจายทั่วไป	ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล	
C=25			
ครั้งที่ 1	0.18	0.18	✓
ครั้งที่ 2	0.18	0.17	✓
ครั้งที่ 3	0.18	0.16	✓
ครั้งที่ 4	0.18	0.18	✓
เฉลี่ย	0.18	0.1725	
C=50			
ครั้งที่ 1	0.47	0.47	✓
ครั้งที่ 2	0.48	0.49	✓
ครั้งที่ 3	0.47	0.47	✓
ครั้งที่ 4	0.48	0.46	✓
เฉลี่ย	0.475	0.4725	
C=150			
ครั้งที่ 1	2.63	2.64	✓
ครั้งที่ 2	2.67	2.66	✓
ครั้งที่ 3	2.65	2.66	✓
ครั้งที่ 4	2.64	2.63	✓
เฉลี่ย	2.6475	2.6475	
C=1,000			
ครั้งที่ 1	39.08	38.37	✓
ครั้งที่ 2	41.78	42.19	✓
ครั้งที่ 3	41.63	40.40	✓
ครั้งที่ 4	47.06	41.05	✓
เฉลี่ย	42.38	40.5025	



ภาพประกอบ 5-4 กราฟแสดงการเปรียบเทียบเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด

จากผลการทดลองที่แสดงดังภาพประกอบ 5-4 จะเห็นได้ว่าดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ใช้เวลาในการค้นหาข้อมูลใกล้เคียงกัน ซึ่งสอดคล้องกับ ตาราง 5-3 ซึ่งแสดงจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบที่ไม่ได้อยู่กลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด

5.2.2.3 ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน

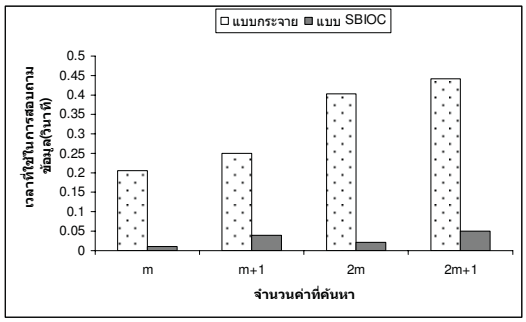
งานวิทยานิพนธ์นี้ ได้ทำการทดลองค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน ดำเนินการโดยการรันโปรแกรมเพื่อสอบถามข้อมูลโดยการแยกพิจรณาค่า m ต่างๆ และ คาร์ดินอลิตี้มีค่าเท่ากับ 25 50 150 และ 1,000 ตามลำดับ แล้วบันทึกเวลาที่ใช้ ซึ่งดำเนินการทั้งหมด 4 ครั้ง (ค่า m ต่างๆ และ คาร์ดินอลิตี้มีค่า m ค่าต่างๆ) จากนั้นหาค่าเฉลี่ยสำหรับการสอบถามข้อมูล ดังผลการทดลองในตาราง 5-7

จากตาราง 5-7 นำค่าที่ได้ที่ใช้ในการสอบถามมาเขียนกราฟได้ดังภาพประกอบ 5-5

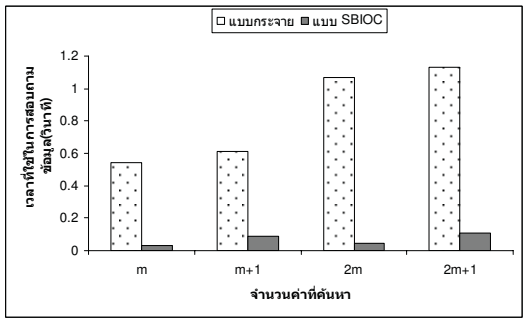
ตาราง 5-7 เวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันบนดัชนีบิตแมปทั้ง 2 ชนิด เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 25, 50, 150 และ 1,000 โดยการแยกพิจารณา ค่า m ต่าง ๆ

คาร์ดินอลิตี้ (C) ค่า m ต่าง ๆ	เวลาที่ใช้ในการสอบถาม (วินาที)		ความถูกต้องของคำตอบ
	ดัชนีบิตแมปแบบกระจายทั่วไป (เฉลี่ย)	ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (เฉลี่ย)	
C=25			
m	0.205	0.01	✓
m+1	0.25	0.04	✓
2m	0.4025	0.02	✓
2m+1	0.4425	0.05	✓
C=50			
m	0.545	0.03	✓
m+1	0.25	0.04	✓
2m	0.4025	0.02	✓
2m+1	0.4425	0.05	✓
C=150			
m	2.8725	0.115	✓
m+1	3.1425	0.325	✓
2m	5.625	0.16	✓
2m+1	5.655	0.365	✓
C=1,000			
m	45.32	0.695	✓
m+1	45.435	2.065	✓
2m	85.775	1.21	✓
2m+1	96.185	2.4325	✓

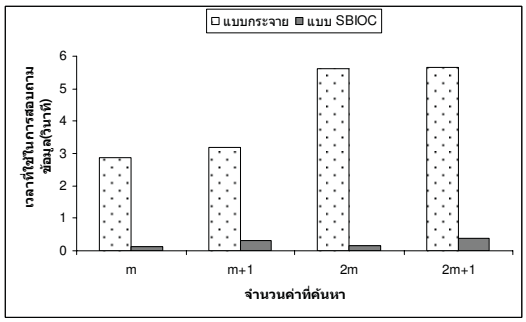
ภาพประกอบ 5-5 เป็นการแสดงการเปรียบเทียบเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด โดยภาพประกอบ 5-5(ก), 5-5(ข), 5-5(ค) และ 5-5(ง) เป็นการค้นหาในกรณีที่จำนวนค่าคาร์ดินอลิตี้ที่ต้องการค้นหาเท่ากับ 25 50 150 และ 1,000 ตามลำดับ



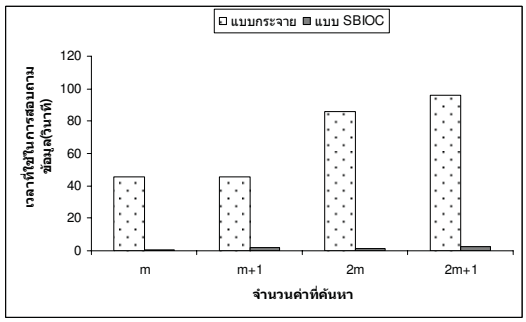
(ก) แผนภาพแสดงเวลาที่ใช้ในการสอบถาม
เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 25



(ข) แผนภาพแสดงเวลาที่ใช้ในการสอบถาม
เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 50



(ค) แผนภาพแสดงเวลาที่ใช้ในการสอบถาม
เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 150



(ง) แผนภาพแสดงเวลาที่ใช้ในการสอบถาม
เมื่อคาร์ดินอลิตี้มีค่าเท่ากับ 1,000

ภาพประกอบ 5-5 กราฟแสดงการเปรียบเทียบเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด

จากผลการทดลองที่แสดงดังภาพประกอบ 5-5 จะเห็นได้ว่าดัชนีบิตแมปแบบกระจายทั่วไป และดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ใช้เวลาในการค้นหาข้อมูลต่างกันมาก กล่าวคือ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูลจะใช้เวลาน้อยกว่า ดัชนีบิตแมปแบบกระจายทั่วไป โดยเฉพาะกรณีที่การค้นหาข้อมูลมีค่าที่เท่ากับจำนวนเท่าของค่า m (จากภาพประกอบ 5-5 กรณีที่ค่าที่ค้นหามีค่า m และ 2m) ซึ่งสอดคล้องกับตาราง 5-4 ซึ่งแสดงจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะระหว่างบิตแมปเวกเตอร์เมื่อมีการสอบถามแบบที่อยู่กลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด

จากผลการทดลองที่แสดงดังภาพประกอบ 5-5 ยังพบอีกว่า ในกรณีที่มีการสอบถามข้อมูลแบบที่อยู่กลุ่มเดียวกัน ยิ่งค่าคาร์ดินอลิตี้มีค่ามากเท่าไร อัตราส่วนเวลาในการค้นหาข้อมูลของดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้เวลาน้อยลงเท่านั้นเมื่อเปรียบเทียบกับดัชนีบิตแมปแบบกระจายทั่วไป

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

คลังข้อมูล เป็นฐานข้อมูลที่จัดเก็บข้อมูลทั้งในอดีตและปัจจุบัน ซึ่งเก็บรวบรวมข้อมูลจากฐานข้อมูลประจำวัน และแหล่งข้อมูลจากภายนอก ข้อมูลอาจนำมาจากส่วนต่าง ๆ ขององค์กร ซึ่งจะถูกเปลี่ยนให้อยู่ในรูปแบบที่มีมาตรฐานเดียวกันทั้งหมด เพื่อให้สามารถนำไปใช้ในการวิเคราะห์ หรือ ช่วยการตัดสินใจให้ผู้บริหารได้ การดึงข้อมูลจากคลังข้อมูลขึ้นมาใช้งานแต่ละครั้ง ส่วนมากมักจะเป็นลักษณะของการสอบถามข้อมูลแบบซับซ้อน (Complex Query) กล่าวคือจะไม่ทราบว่าผู้ใช้จะสอบถามข้อมูลอะไรบ้าง จึงต้องมีการรวบรวมข้อมูลจากหลายแหล่งมาไว้ที่ศูนย์กลาง และเก็บบันทึกทั้งข้อมูลในอดีตและปัจจุบันไว้ในคลังข้อมูล ขอบเขตของข้อมูลที่เก็บไว้จึงกว้างขวาง ครอบคลุมทุกแง่มุมที่สำคัญขององค์กรนั้น ๆ เพื่อให้พร้อมใช้งานในการสอบถามข้อมูลทุกรูปแบบ ดังนั้นคลังข้อมูลจะมีขนาดใหญ่ ทำให้การค้นหาข้อมูลย่อมใช้เวลามากขึ้น

วิธีการหนึ่ง ในการลดเวลาในการค้นหาข้อมูล (เพิ่มประสิทธิภาพ) โดยไม่ต้องเสียค่าใช้จ่ายในการเพิ่มอุปกรณ์ฮาร์ดแวร์ คือ การทำดัชนีแบบบิตแมป โดยดัชนีแบบบิตแมปเหมาะกับข้อมูลที่แอทริบิวต์มีค่าคาร์ดินอลิตี้ต่ำ (จำนวนค่าที่แตกต่างกันบนคอลัมน์ที่นำมาทำดัชนีมีค่าแตกต่างไม่มากนัก) เช่น เพศ จังหวัด เป็นต้น และเหมาะกับข้อมูลที่ไม่เปลี่ยนแปลง ซึ่งพบบ่อยในคลังข้อมูล

ได้มีงานวิจัยที่เกี่ยวกับการทำดัชนีแบบบิตแมป ในคลังข้อมูลมากมาย ซึ่งงานวิจัยแต่ละชิ้นมุ่งเน้นในการลดพื้นที่ในการจัดเก็บ และเพิ่มความเร็วในการค้นหาข้อมูล หรือทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล และได้มีงานวิจัยที่ยังแยกประเภทของการค้นหาข้อมูล (query) เข้ามาเกี่ยวข้อง ซึ่งแยกเป็นการค้นหาแบบเท่ากันและค้นหาแบบสมาชิก เช่น ดัชนีบิตแมปแบบพื้นฐาน เป็นดัชนีที่มีความเร็วในการค้นหาข้อมูลแบบเท่ากัน แต่ไม่เหมาะสำหรับแอทริบิวต์สูง ๆ เพราะใช้พื้นที่จัดเก็บมาก ดัชนีบิตแมปแบบเข้ารหัส ซึ่งเป็นดัชนีบิตแมปที่เน้นการลดพื้นที่การจัดเก็บแต่ใช้เวลานานในการค้นหาข้อมูล ดัชนีบิตแมปแบบกระจาย จะรวมทั้งลดพื้นที่และเพิ่มความเร็วในการค้นหาข้อมูล เป็นต้น นอกจากนี้เรายังพบอีกว่าจากโครงสร้างของดัชนีบิตแมปแบบกระจาย เรายังสามารถเพิ่มความเร็วในการค้นหาข้อมูลได้อีก โดยการพิจารณาการค้นหาข้อมูลที่อยู่ในกลุ่มสมาชิกเดียวกัน

ตาราง 6-1 ลักษณะที่สำคัญของดัชนีบิตแมปทั้ง 2 ชนิด (C คือ จำนวนค่าที่เป็นไปได้ของแอทริบิวต์ที่เลือกมาทำดัชนี, N คือ จำนวนเรคอร์ด)

การพิจารณา	ชนิดของดัชนีบิตแมป	
	ดัชนีบิตแมปแบบกระจายทั่วไป	ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล
พื้นที่ที่ใช้ในการจัดเก็บดัชนี	$\lceil 2\sqrt{C} \rceil N$	$\lceil 2\sqrt{C} \rceil N$
ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน	2:1 (1AND)	2:1 (1AND)
ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน	2n:2n-1 (n AND, n-1 OR)	2n:2n-1 (n AND, n-1 OR)
ค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกัน	2n:2n-1 (n AND, n-1 OR)	1:0

ในงานวิทยานิพนธ์นี้ ได้เสนอเทคนิคการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้นโดยใช้การแบ่งกลุ่มข้อมูล (Data Clustering) ซึ่งเป็นหนึ่งในเทคนิคการทำเหมืองข้อมูล (Data Mining) มาช่วยในการจัดกลุ่มและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจาย ซึ่งสามารถพิจารณาการเปรียบเทียบลักษณะที่สำคัญของดัชนีบิตแมปแบบกระจายทั่วไป และ ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล (SBIOC) ได้ดังตาราง 6-1

จากตาราง 6-1 จะเห็นได้ว่า ดัชนีบิตแมปทั้ง 2 ชนิดมีลักษณะโครงสร้างที่เหมือนกันแต่ที่มาจะแตกต่างกัน ส่งผลให้มีข้อดีข้อจำกัดที่แตกต่างกันไป ซึ่งดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล ถูกสร้างขึ้นมาเพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้น เมื่อพิจารณาการสอบถามแบบต่าง ๆ พบว่า ดัชนีบิตแมปทั้ง 2 ชนิดมีลักษณะที่เป็นจุดเด่นและจุดด้อย ดังนี้

- กรณีพิจารณาพื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปของทั้ง 2 ชนิด พบว่าจะใช้พื้นที่ในการจัดเก็บเท่ากัน กล่าวคือ ดัชนีบิตแมปทั้ง 2 ชนิด ใช้พื้นที่ในการจัดเก็บ $2\sqrt{C}$ บิตแมปเวกเตอร์

- กรณีพิจารณาค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปทั้ง 2 ชนิด พบว่าค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันจะมีค่าใช้จ่ายเท่ากัน กล่าวคือ ดัชนีบิตแมปทั้ง 2 ชนิดมีค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันเป็น 2:1 (1AND) กล่าวคือ อ่านข้อมูล 2 ครั้ง และดำเนินการตรรกะ AND 1 ครั้ง

- กรณีพิจารณาค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด พบว่าค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกันจะมีค่าใช้จ่ายเท่ากัน กล่าวคือ ดัชนีบิตแมปทั้ง 2 ชนิดมีค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่ไม่ได้อยู่ในกลุ่มเดียวกัน เป็น 2n:2n−1 (n AND, n−1 OR) กล่าวคือ อ่านข้อมูล 2n ครั้ง และดำเนินการตรรกะ 2n−1 ครั้ง

- กรณีพิจารณาค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันของดัชนีบิตแมปทั้ง 2 ชนิด พบว่าค่าใช้จ่ายเกี่ยวกับเวลาที่ใช้ในการสอบถามแบบสมาชิกที่อยู่ในกลุ่มเดียวกันจะมีค่าใช้จ่ายต่างกัน กล่าวคือ ดัชนีบิตแมปแบบกระจายทั่วไปมีค่าใช้จ่ายเป็น 2n:2n−1 (n AND, n−1 OR) กล่าวคือ อ่านข้อมูล 2n ครั้ง และดำเนินการตรรกะ 2n−1 ครั้ง แต่ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล จะใช้เพียง 1:0 เท่านั้น

การเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายให้มากยิ่งขึ้นโดยใช้การแบ่งกลุ่มข้อมูล ซึ่งเป็นหนึ่งในเทคนิคการทำเหมืองข้อมูลมาช่วยในการจัดกลุ่มและแบ่งแยกกลุ่มข้อมูลอย่างชัดเจน เพื่อการเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบสมาชิกของดัชนีบิตแมปแบบกระจายทั่วไป จากที่กล่าวมาทั้งหมด จะเห็นได้ว่า ดัชนีบิตแมปทั้ง 2 ชนิดมีข้อดี และข้อจำกัดที่แตกต่างกันไป การนำไปใช้จึงควรเลือกให้เหมาะสมกับความต้องการ

6.2 ข้อเสนอแนะและงานในอนาคต

ดัชนีบิตแมปแบบกระจายด้วยเทคนิคการแบ่งกลุ่มข้อมูล มีข้อดีในเรื่องของการประหยัดพื้นที่ในการจัดเก็บดัชนี และประหยัดเวลาในการค้นหาข้อมูล กล่าวคือ เหมาะสำหรับการสอบถามข้อมูลแบบความเป็นสมาชิกกลุ่มเดียวกัน โดยเฉพาะอย่างยิ่งจำนวนค่าที่ต้องการค้นหา เท่ากับ จำนวนเท่าของสมาชิกในกลุ่ม Z (ค่า m) สำหรับข้อเสนอแนะและงานวิจัยที่น่าสนใจที่ควรทำในอนาคตมีดังนี้

- การนำเทคนิคหลาย ๆเทคนิคของการทำเหมืองข้อมูลมาผสมประสานกัน อาจเป็นทางเลือกหนึ่งในการช่วยหาความสัมพันธ์ของข้อมูลในการรวมกลุ่มของข้อมูลกรณีที่มีสมาชิกของกลุ่ม มีจำนวนมากกว่าจำนวนเท่าของสมาชิกในกลุ่ม Z (ค่า m)

- งานวิจัยที่น่าสนใจที่ควรทำในอนาคต คือ เทคนิคการค้นหาแบบซ้อนกันเป็นชั้น ๆ ของการค้นหาข้อมูลในดัชนีบิตแมปแบบกระจาย เพื่อรองรับการค้นหาข้อมูลในแอทริบิวต์ที่

มีค่าคาร์ดินอลิตี้สูง ๆ โดยอาจจะใช้เทคนิคของการทำเหมืองข้อมูล หลาย ๆ เทคนิค มาประยุกต์ใช้งานร่วมกัน

- งานวิจัยที่น่าสนใจที่ควรทำต่อเนื่องกัน คือ คิดค้นการทำดัชนีบิตแมปแบบกระจาย ในการค้นหาข้อมูลที่มีการค้นหาข้อมูลในหลาย ๆ แอทริบิวต์ พร้อม ๆกัน

บรรณานุกรม

[1] A. Silberschatx , H.F .Korth, and S.Sudarshan, "Database System Concept," Mc Graw Hill, 2001.

[2] C.Y. Chan and Y. E. Ioannidis, "An Efficient Bitmap Encoding Scheme for Selection Queries," Proceeding of the 1999 ACM SIGMOD 1999.

[3] S. Vanichayobon, J. Manfuekphan and L. Gruenwald, "Scatter Bitmap: Space-Time Efficient Bitmap Indexing for Equality and Membership Queries," Proceedings of IEEE International Conferences on Cybernetics and Intelligent Systems (CIS 2006).

[4] M. Wu and A. P. Buchmann, "Encoded Bitmap Indexing for Data Warehouses," Proceedings of the Fourteenth International Conference on Data Engineering, 1998.

[5] A. Berson and S. J. Smith, "Data Warehousing and Data Mining," McGraw -Hill, 1977.

[6] J. Han and M. Kamber, "Data Warehouses and OLAP Technology for Data Mining," 2000.

[7] J. Han and M. Kamber, "Data Mining: Concepts and Techniques." Morgan-Kaufmann, San Francisco, 2001.

[8] R. Kimball, "Data Warehouse Designer," Intelligent Enterprise Magazine, 2008.

[9] M. Zaman, J. Surabattula, and L. Gruenwald, "An Auto-Indexing Technique for Databases Based on Clustering," Proceedings of the Database and Expert Systems Applications, 15th International Workshop on (DEXA'04), 2004.

[10] V. Rapševičius, A. Juozapavičius and A. Brazauskas, "Clustering of Descriptive-textual Data on Silurian Rocks of Lithuania," Geologija. Vilnius. 2006.

[11] J. Manfuekphan, S. Vanichayobon, and D. Salee, "Scatter Bitmap Index Optimization by using Association Rules," The Joint Conference Computer Science and Software Engineering (JCSSE 2006), 2006.

[12] R. Raghu and G. Johannes, "Database Management Systems," 2000.

[13] N. Svetlozar and J. Nenad, "Ad-Hoc Association-Rule Mining within the Data Warehouse. Proceedings of the 36th Hawaii International Conference on System Sciences(HICSS'03)," IEEE, [On-line]. Available: http://csdl.computer.org/comp/proceedings/hicss/2003/1874/08/187480232a.pdf. [Access date : July 1, 2008].

[14] R. Vilin, "Bitmap Indexing In Oracle8 Data Warehousing," [On-line]. Available : http://www.nocoug.org/download/2003-11/BMI-NoCOUG.org-Nov19-03.ppt. [Access date : July 1, 2008].

[15] "Searching (Articles) ," [On-line]. Available: http://cptd.chandra.ac.th/selfstud/search.html. [Access date : July 1, 2008].

[16] "Indexing (Articles)." [On-line]. Available: http://ced.kmitnb.ac.th/tawa /Indexing-sent.doc. [Access date : July 1, 2008].

[17] "Data mining (Articles)." [On-line]. Available: http://std.eng.src.ku.ac.th/CPE47 /subject/Data_Mining/video/?C=N;O=D. [Access date : July 1, 2008].

[18] "Transaction Processing Performance Council (TPC). TPC-H: An Ad-hoc, Decision Support Benchmark. Standard Specification Revision 2.1.0." [On-line]. Available: http://www.tpc.org/tpch/default.asp. [Access date : Mar 1, 2008].

[19] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," ACM SIGMOD Record, Vol. 26(1), March 1997, pp.65-74.

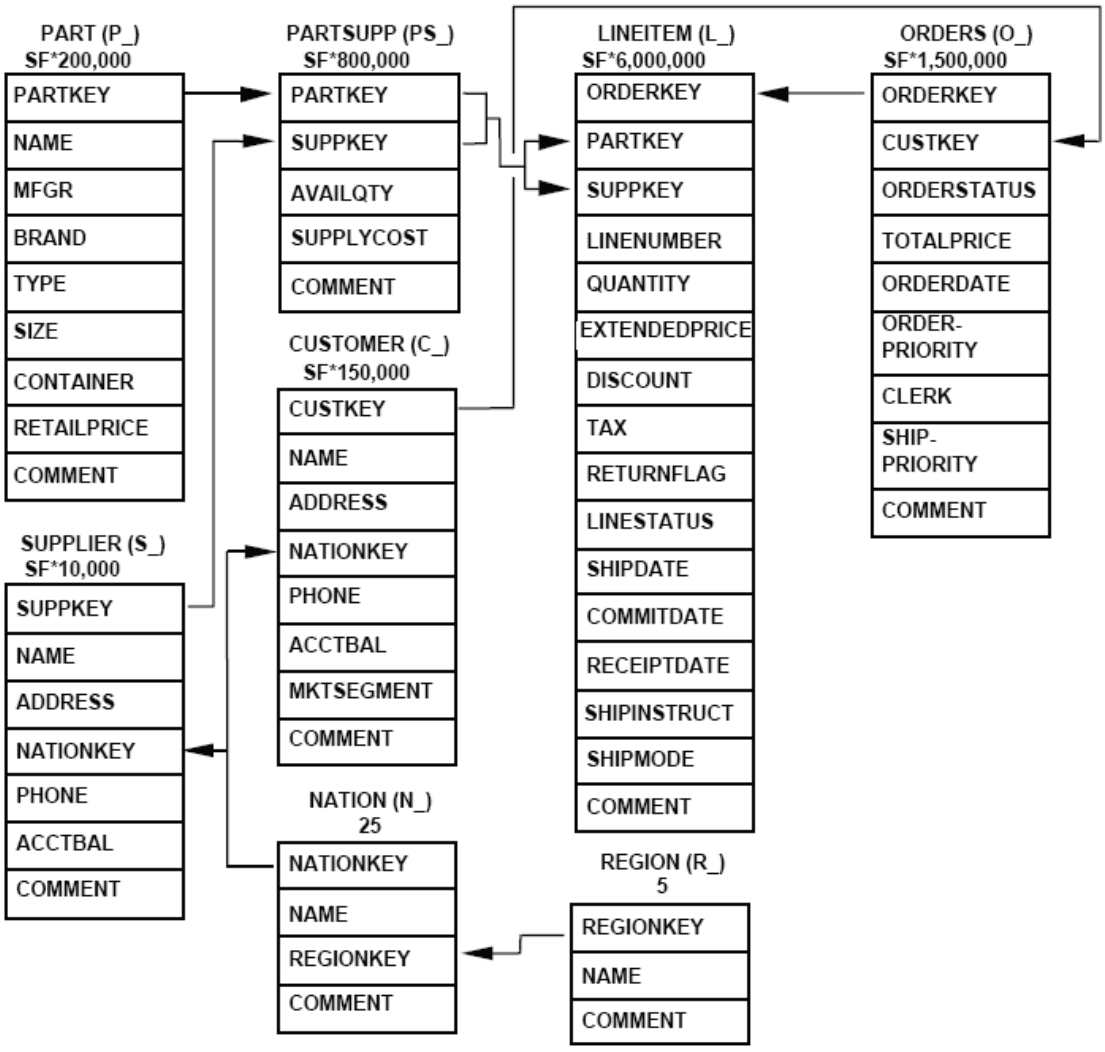
[20] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms," USA: MIT Press, 2003.

ภาคผนวก

ภาคผนวก

ก. การวัดเปรียบเทียบสมรรถนะด้วย TPC-H

ข้อมูลที่ใช้ในการทดลองในวิจัยชิ้นนี้ เป็นข้อมูลมาตรฐานจาก TPC-H Benchmark ซึ่งเป็นตัววัดเปรียบเทียบสมรรถนะที่ใช้ในระบบสนับสนุนการตัดสินใจ (Decision Support) โดยต้องการทดสอบกับข้อมูลที่มีปริมาณมาก ประมวลผลการสอบถามที่เป็นแบบซับซ้อน โดยมีโครงสร้าง ดังนี้



ภาพประกอบ ก-1 โครงสร้าง TPC-H

ก.1 ข้อมูลตารางและแอทริบิวต์ที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองมี 4 ชุด และมาจาก 2 ตาราง กล่าวคือ ข้อมูลจากตาราง PART และ ORDER ซึ่งมีรายละเอียดดังนี้

ข้อมูลชุดที่ 1 แอทริบิวต์ BRAND บน PART Table
มี 200,000 แถว คาร์ดินอลิตี้เท่ากับ 25 (C = 25)

ข้อมูลชุดที่ 2 แอทริบิวต์ SIZE บน PART Table
มี 200,000 แถว คาร์ดินอลิตี้เท่ากับ 50 (C = 50)

ข้อมูลชุดที่ 3 แอทริบิวต์ TYPE บน PART Table
มี 200,000 แถว คาร์ดินอลิตี้เท่ากับ 150 (C = 150)

ข้อมูลชุดที่ 4 แอทริบิวต์ CLERK บน ORDER Table
มี 1,500,000 แถว คาร์ดินอลิตี้เท่ากับ 1,000 (C = 1,000)

ก.2 โครงสร้างตาราง (Table Layouts) ของการวัดเปรียบเทียบสมรรถนะ

ฐานข้อมูลของการวัดเปรียบเทียบสมรรถนะของ TPC-H จะประกอบด้วยตารางเชิงความสัมพันธ์หลายตาราง เช่น ตาราง CUSTOMER จะเป็นตารางเชิงความสัมพันธ์ที่เก็บข้อมูลเกี่ยวกับลูกค้า ตาราง PART จะเป็นตารางเชิงความสัมพันธ์ที่เก็บข้อมูลเกี่ยวกับชิ้นส่วนของสินค้า โดยมีโครงสร้างตารางดังนี้

โครงสร้างตาราง PART

ชื่อแอทริบิวต์	**ชนิดข้อมูล**	**หมายเหตุ**
P_PARTKEY	identifier	SF*200,000 are populated
P_NAME	variable text, size 55	
P_MFGR	fixed text, size 25	
P_BRAND	fixed text, size 10	
P_TYPE	variable text, size 25	
P_SIZE	integer	
P_CONTAINER	fixed text, size 10	
P_RETAILPRICE	decimal	
P_COMMENT	variable text, size 23	

คีย์หลัก (Primary Key): P_PARTKEY

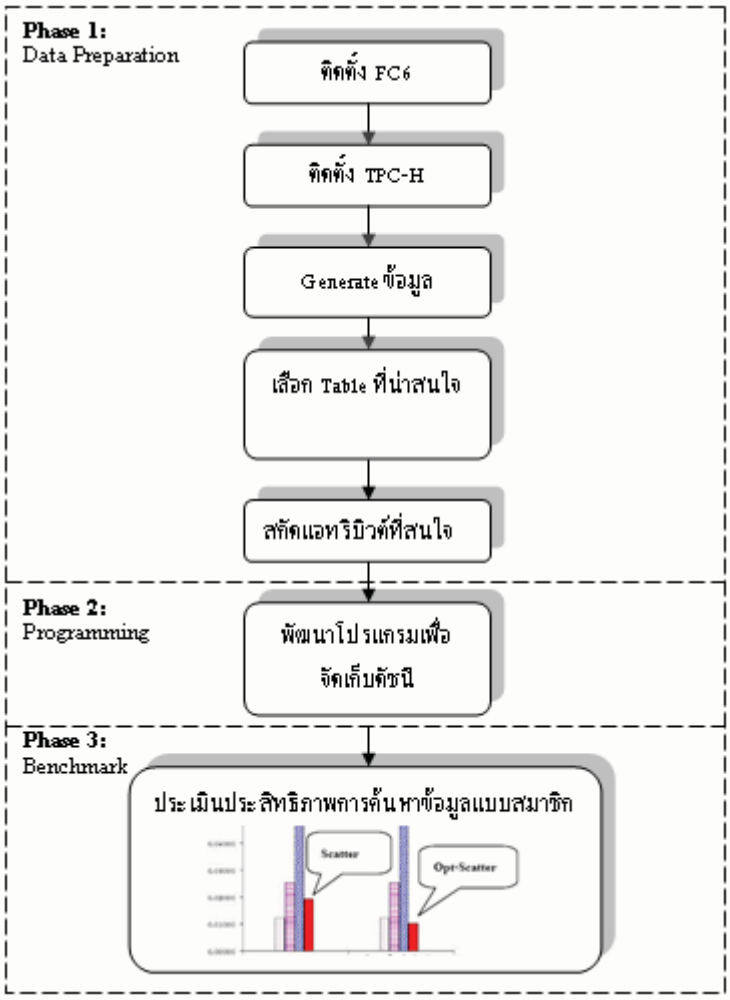
โครงสร้างตาราง ORDERS

ชื่อแอทริบิวต์	**ชนิดข้อมูล**	**หมายเหตุ**
O_ORDERKEY	identifier	SF*1,500,000 are sparsely populated
O_CUSTKEY	identifier	Foreign key reference to C_CUSTKEY
O_ORDERSTATUS	fixed text, size 1	
O_TOTALPRICE	decimal	
O_ORDERDATE	date	
O_ORDERPRIORITY	fixed text, size 15	
O_CLERK	fixed text, size 15	
O_SHIPPRIORITY	integer	
O_COMMENT	variable text, size 79	

คีย์หลัก (Primary Key): O_ORDERKEY

ข. โครงสร้างขั้นตอนการทดลองและแผนผัง

ส่วนนี้เรานำเสนอ ขั้นตอนการทดลองพัฒนาดัชนีการค้นหา เพื่อการวัดเปรียบเทียบสมรรถนะทางด้านเวลาในการค้นหาของดัชนีบิตแมปทั้งสอง โดยโครงสร้างของการทดลองดัชนีการค้นหาข้อมูลจะแบ่งขั้นตอนการทำงานออกเป็น 3 ส่วนหลัก กล่าวคือ การเตรียมข้อมูล การพัฒนาอัลกอริทึมในการจัดเก็บ และ วัดเปรียบเทียบสมรรถนะทางด้านเวลาในการค้นหาของดัชนีบิตแมปทั้งสอง ซึ่งแต่ละส่วนมีรายละเอียดดังนี้



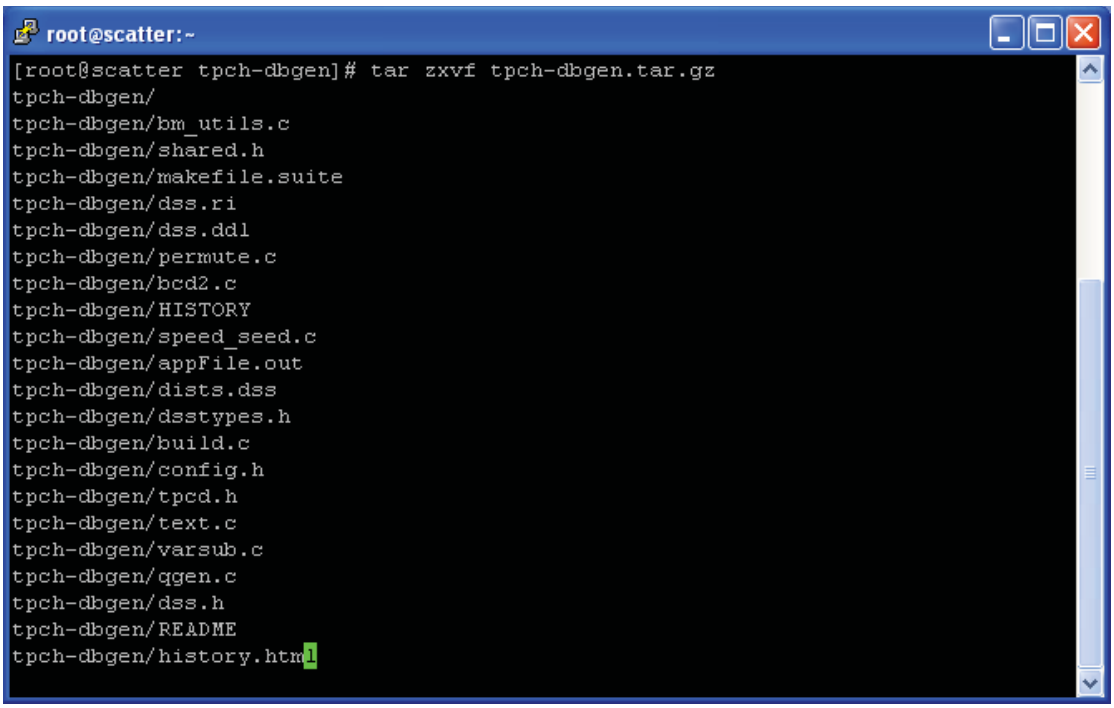
ภาพประกอบ ข-1 โครงสร้างขั้นตอนการทดลอง

- **ขั้นตอนที่ 1: การเตรียมข้อมูล (Data Preparation)**

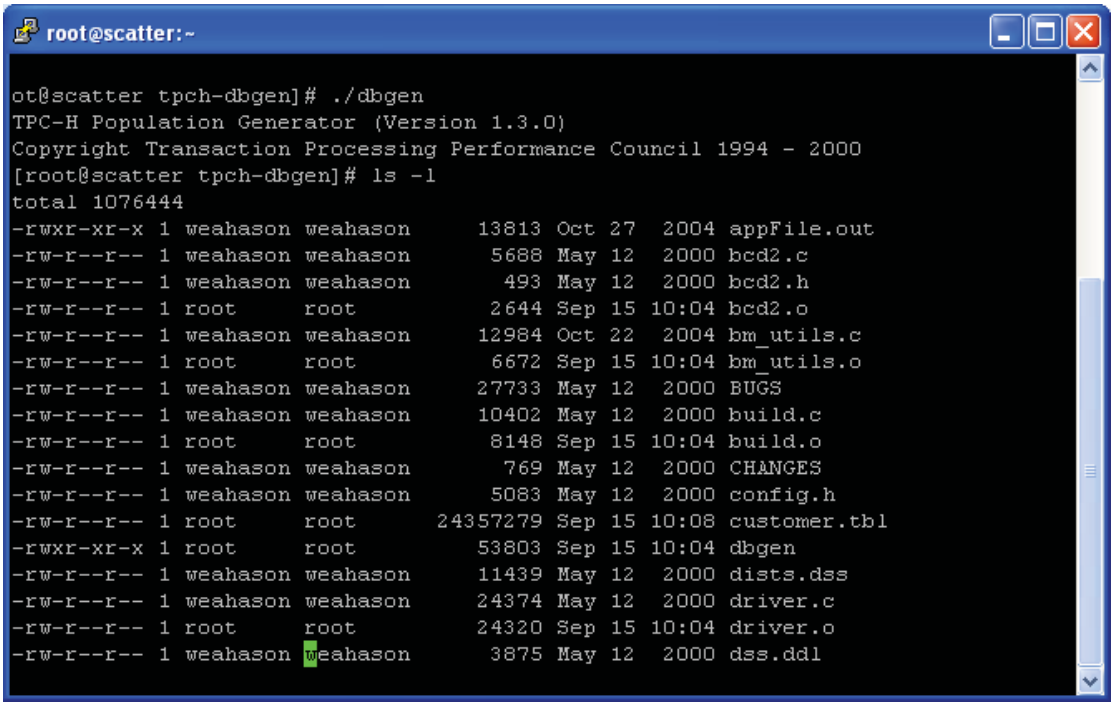
การเตรียมข้อมูลประกอบด้วย 5 ขั้นตอนดังนี้

1. ทำการติดตั้ง Fedora Core 6 เพื่อเป็น Server ในการทดลอง TPC-H Benchmark และติดตั้ง packet ต่างๆที่ใช้ประกอบการทดลองในงานวิจัยชิ้นนี้ ตัวอย่างเช่น gcc, samba, ssh, firewall เป็นต้น

2. Download code TPC-H และทำการติดตั้งเข้าไปในระบบปฏิบัติการ Fedora Core 6 ดังภาพประกอบ ข-2

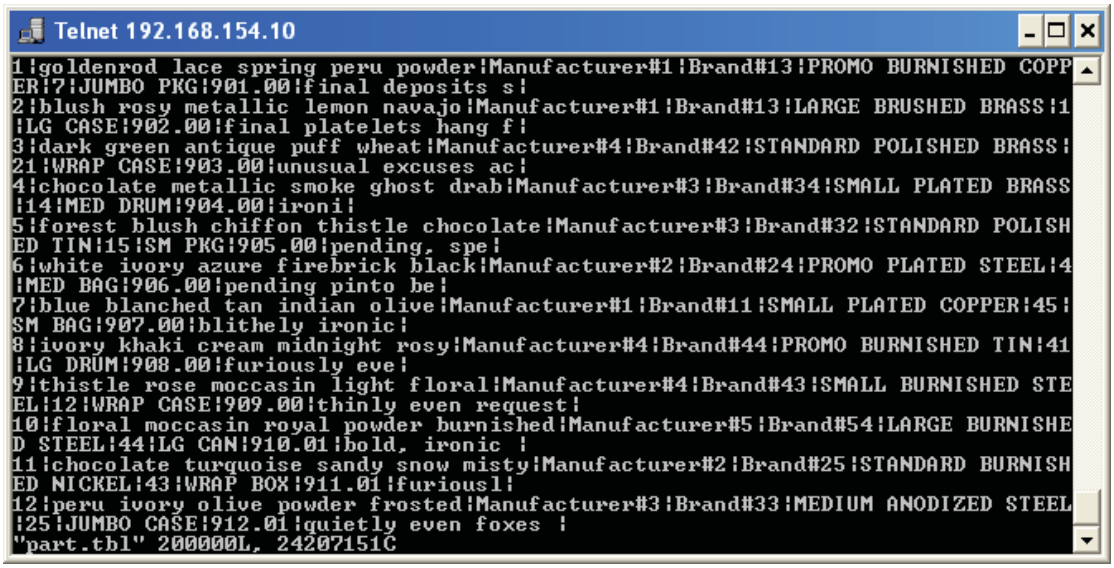


ภาพประกอบ ข-2 ขั้นตอนการติดตั้ง TPC-H



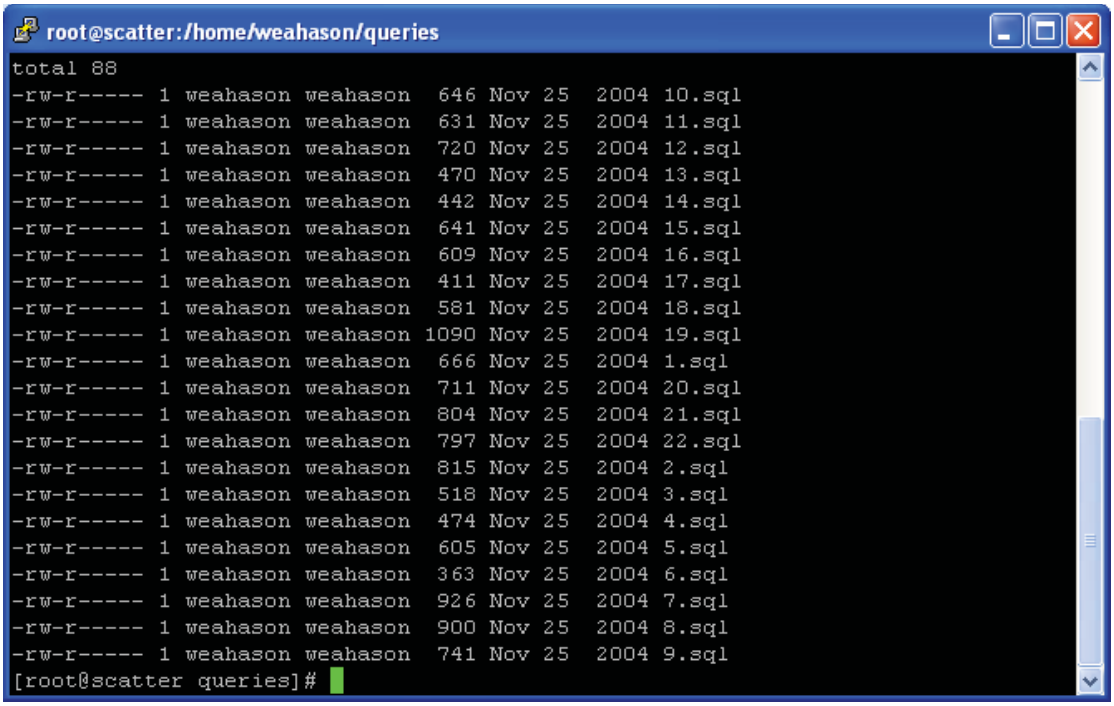
ภาพประกอบ ข-3 วิธีการ generate ข้อมูลดิบจาก TPC-H

3. ทำการ Generate ข้อมูลดิบจาก TPC-H ดังภาพประกอบ ข-3 และหลังจาก Generate ข้อมูลแล้ว จะได้ข้อมูลดิบดังภาพประกอบ ข-4 กล่าวคือ ลักษณะของข้อมูลจะติดกัน แต่จะมีเครื่องหมาย “|” ในการแยกข้อมูล



ภาพประกอบ ข-4 ข้อมูลดิบที่ได้หลังจาก generate ข้อมูล

4. ตรวจสอบ Table ที่น่าสนใจจาก Standard Query ของ TPC-H ซึ่งมีตัวอย่างการ Query ของ TPC-H ที่มีจำนวน 22 Query ดังภาพประกอบ ข-5 และ ภาพประกอบ ข-6 เป็นตัวอย่างคำสั่งการ Query ของ Query ลำดับที่ 19

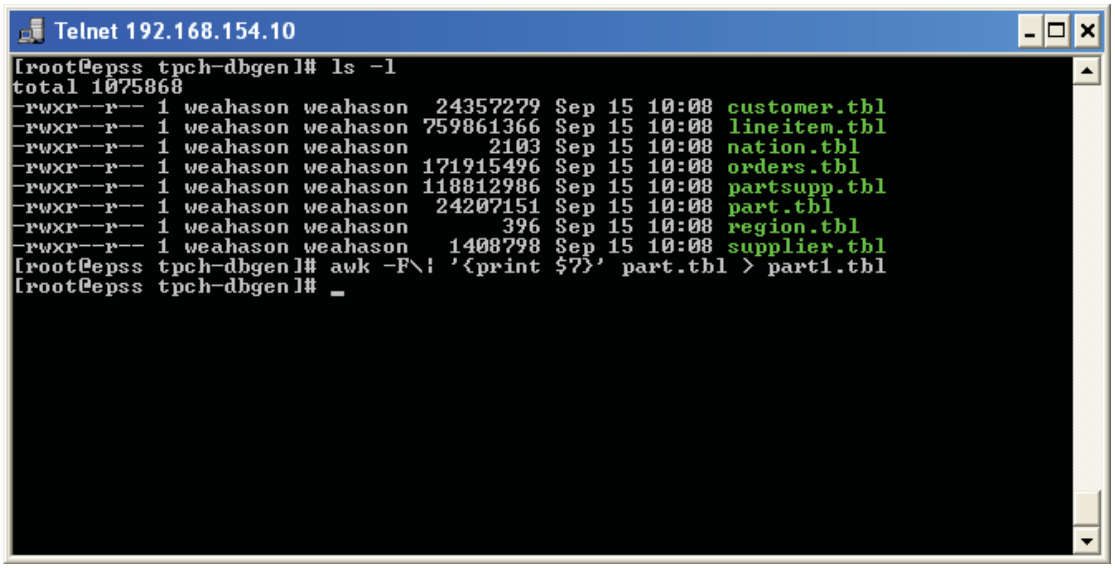


ภาพประกอบ ข-5 แสดงตัวอย่าง Standard Query ของ TPC-H

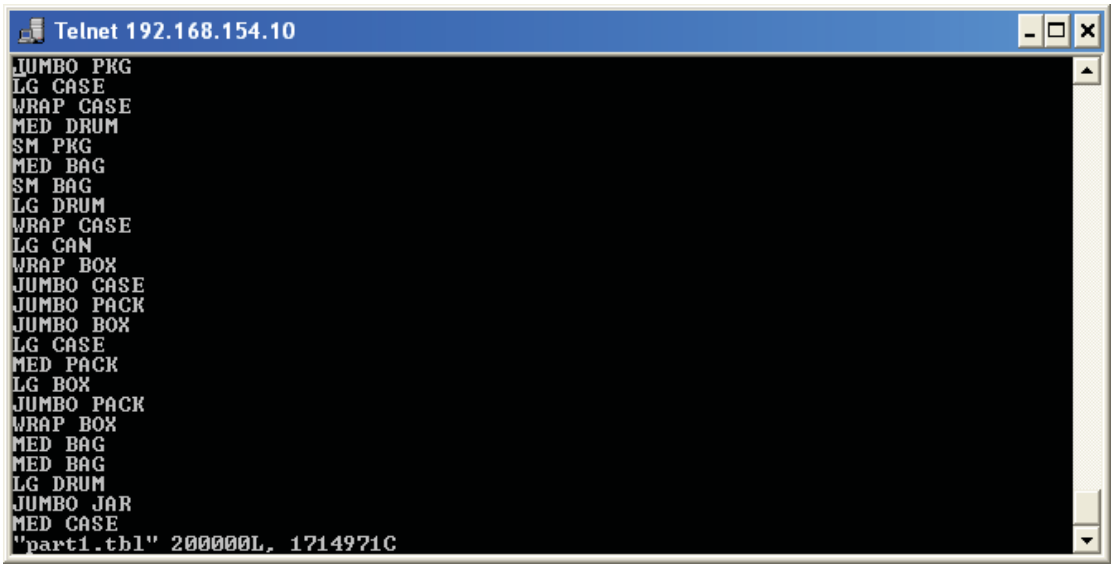
```
-- $ID$
-- TPC-H/TPC-R Discounted Revenue Query (Q19)
-- Functional Query Definition
-- Approved February 1998
:x
:o
select
	sum(l_extendedprice* (1 - l_discount)) as revenue
from
	lineitem,
	part
where
	(
		p_partkey = l_partkey
		and p_brand = ':1'
		and p_container in ('SM CASE', 'SM BOX', 'SM
PACK', 'SM PKG')
		and l_quantity >= :4 and l_quantity <= :4 + 10
		and p_size between 1 and 5
		and l_shipmode in ('AIR', 'AIR REG')
		and l_shipinstruct = 'DELIVER IN PERSON'
	)
	or
	(
		p_partkey = l_partkey
		and p_brand = ':2'
		and p_container in ('MED BAG', 'MED BOX', 'MED
PKG', 'MED PACK')
		and l_quantity >= :5 and l_quantity <= :5 + 10
		and p_size between 1 and 10
		and l_shipmode in ('AIR', 'AIR REG')
		and l_shipinstruct = 'DELIVER IN PERSON'
	)
	or
	(
		p_partkey = l_partkey
		and p_brand = ':3'
		and p_container in ('LG CASE', 'LG BOX', 'LG PACK',
'LG PKG')
		and l_quantity >= :6 and l_quantity <= :6 + 10
		and p_size between 1 and 15
		and l_shipmode in ('AIR', 'AIR REG')
		and l_shipinstruct = 'DELIVER IN PERSON'
	);
```

ภาพประกอบ ข-6 ตัวอย่างคำสั่ง Query ที่ 19 ของ TPC-H

5. ทำการสกัดแอทริบิวต์ที่สนใจ โดยใช้ความสามารถของ OS Fedora Core 6 ดังภาพประกอบ ข-7 และ ภาพประกอบ ข-8 เป็นผลจากการสกัดแอทริบิวต์

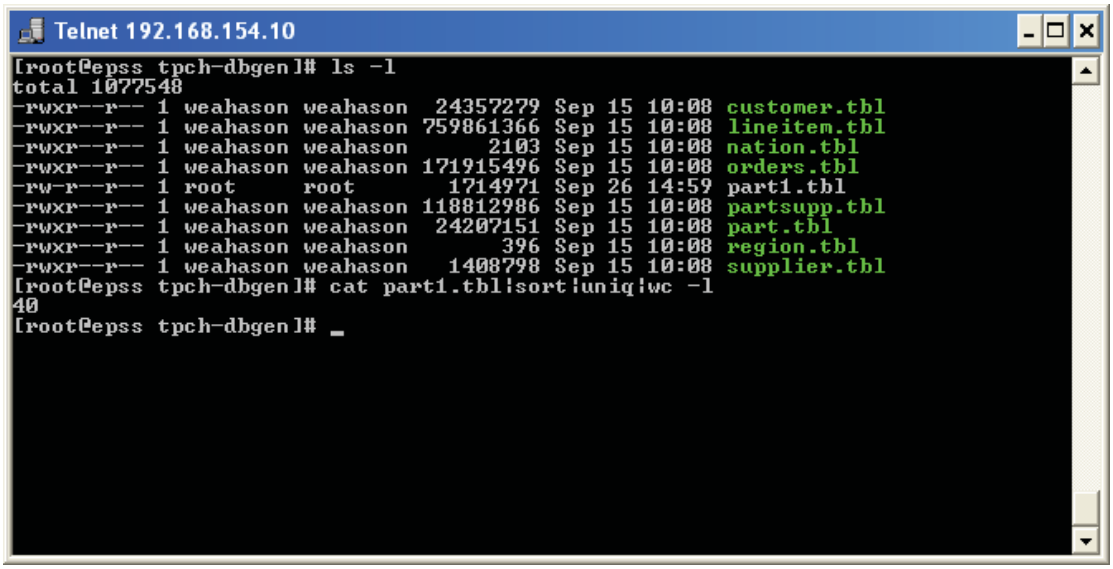


ภาพประกอบ ข-7 วิธีการสกัดแอทริบิวต์ที่ 7 จากข้อมูล Table part.tbl



ภาพประกอบ ข-8 ผลจากการสกัดแอทริบิวต์

6. ทำการตรวจสอบแอทริบิวต์ที่สนใจนั้น มีค่าคาร์ดินอลิตี้เท่าไรโดยใช้ความสามารถของ OS ในการตรวจนับดังภาพประกอบ ข-9 จากภาพประกอบจะพบว่า ข้อมูลใน part1.tbl จะมีค่าคาร์ดินอลิตี้ เท่ากับ 40



ภาพประกอบ ข-9 วิธีการนับคาร์ดินอลิตี้

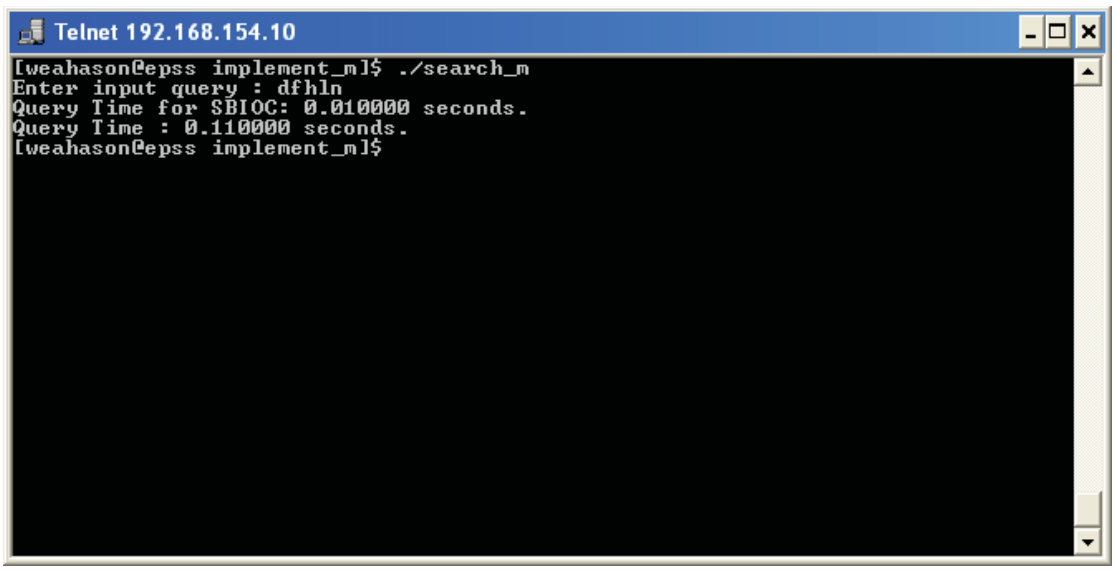
- **ขั้นตอนที่ 2: การเขียนโปรแกรม (Programming)**

ส่วนนี้เป็นขั้นตอนการเขียนโปรแกรมเพื่อทำกาจัดเก็บข้อมูลบิทแมปเวกเตอร์ต่างๆ และการค้นหาข้อมูล โดยใช้ ภาษา C ในการเขียนโปรแกรม

- **ขั้นตอนที่ 3: ประเมินประสิทธิภาพการวัดเปรียบเทียบสมรรถนะ (Benchmark)**

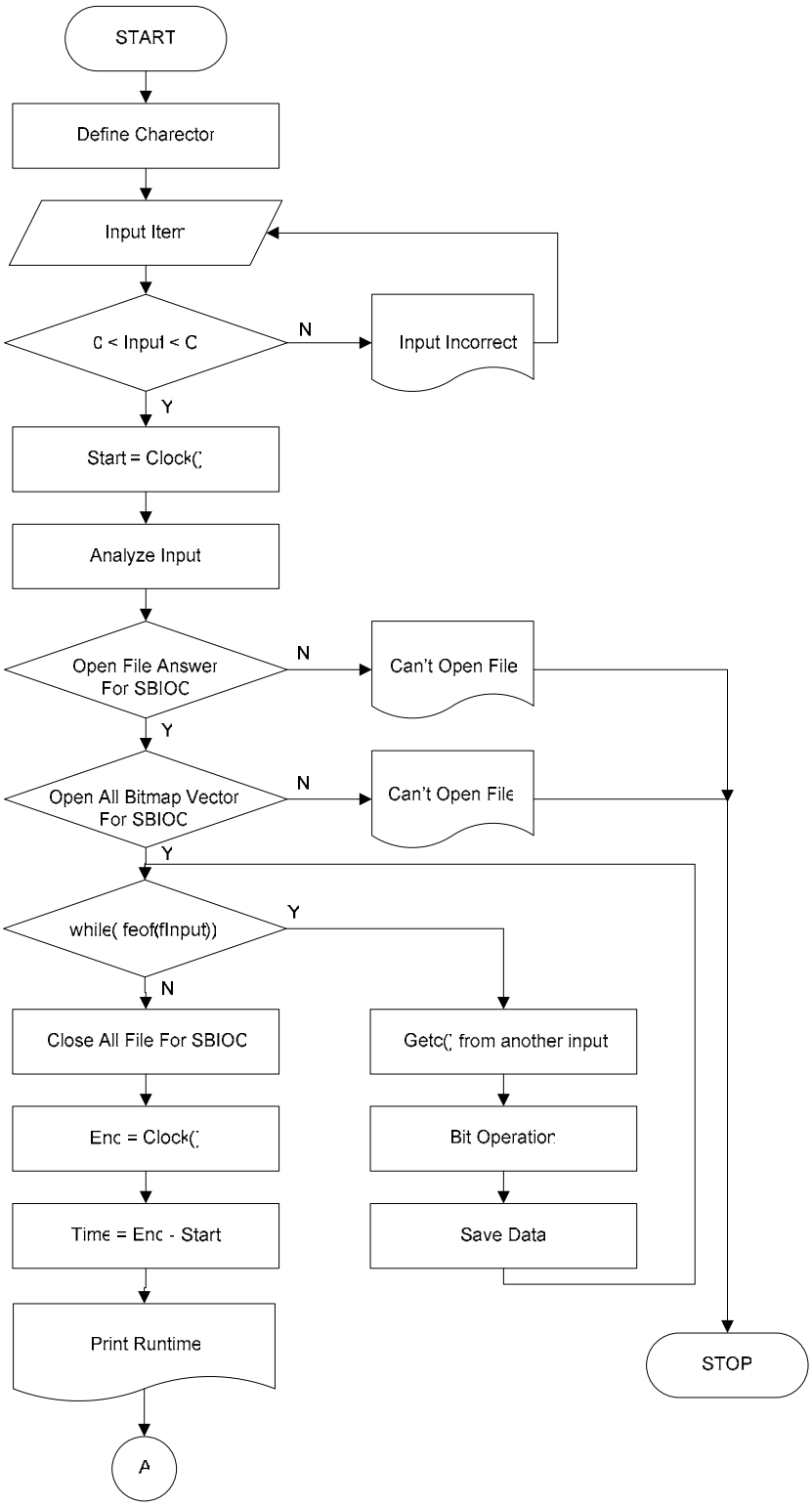
ส่วนนี้เป็นขั้นตอนการวัดเปรียบเทียบสมรรถนะ [18] ในการค้นหาข้อมูลโดยใช้วิธีการจับเวลาในการค้นหา ดังภาพประกอบ ข-10 และ ทำการตรวจสอบความถูกต้องของข้อมูล โดยการจัดเก็บข้อมูลที่ได้จากการค้นหาไว้ในแฟ้มข้อมูล แล้วนำแฟ้มข้อมูลต้นฉบับและแฟ้มข้อมูลที่ได้จากการค้นหามาเปรียบเทียบว่าเหมือนกันหรือไม่ โดยการใช้คำสั่ง diff ใน Linux ดังนี้

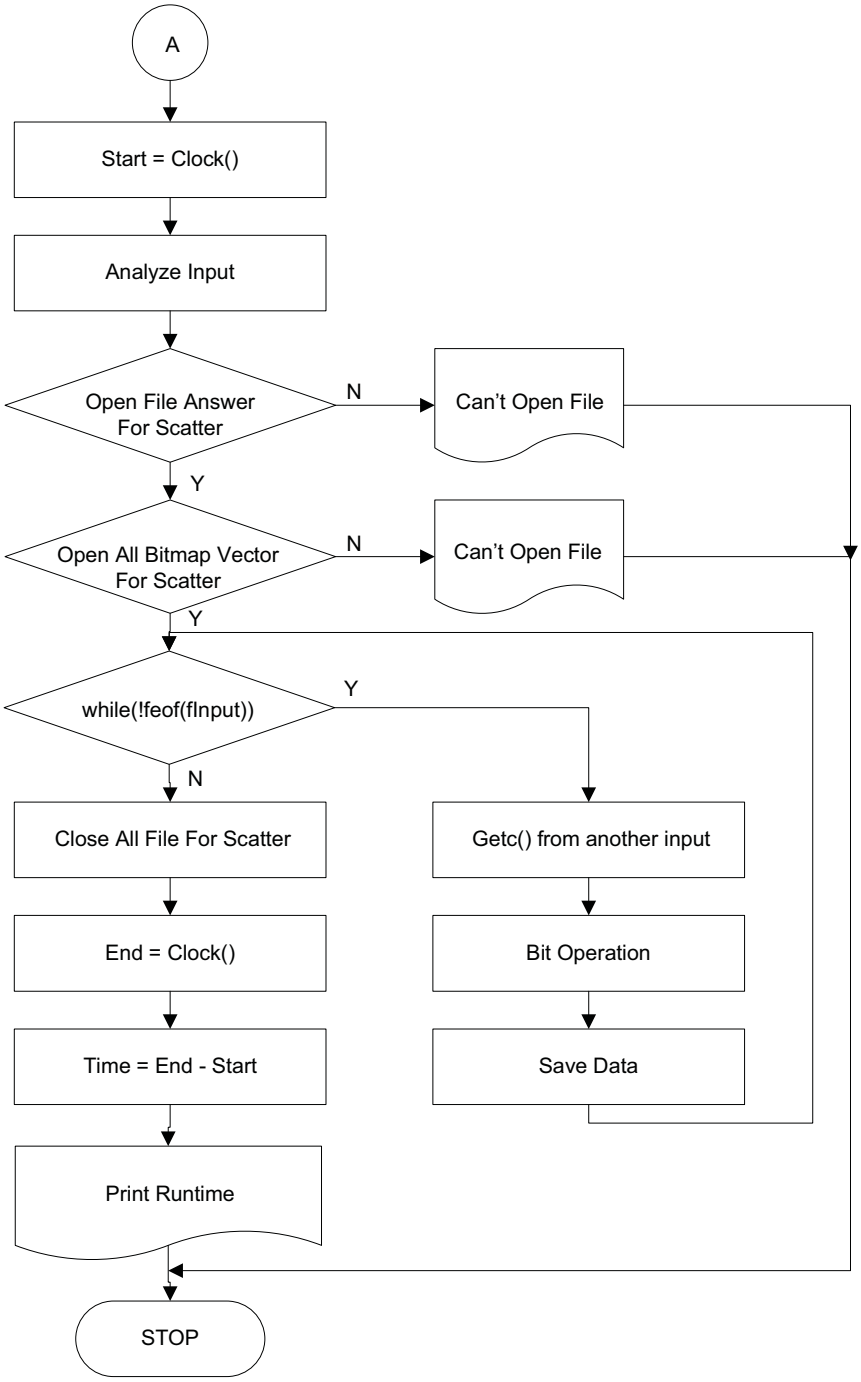
```
diff [option] from-File To-File
```



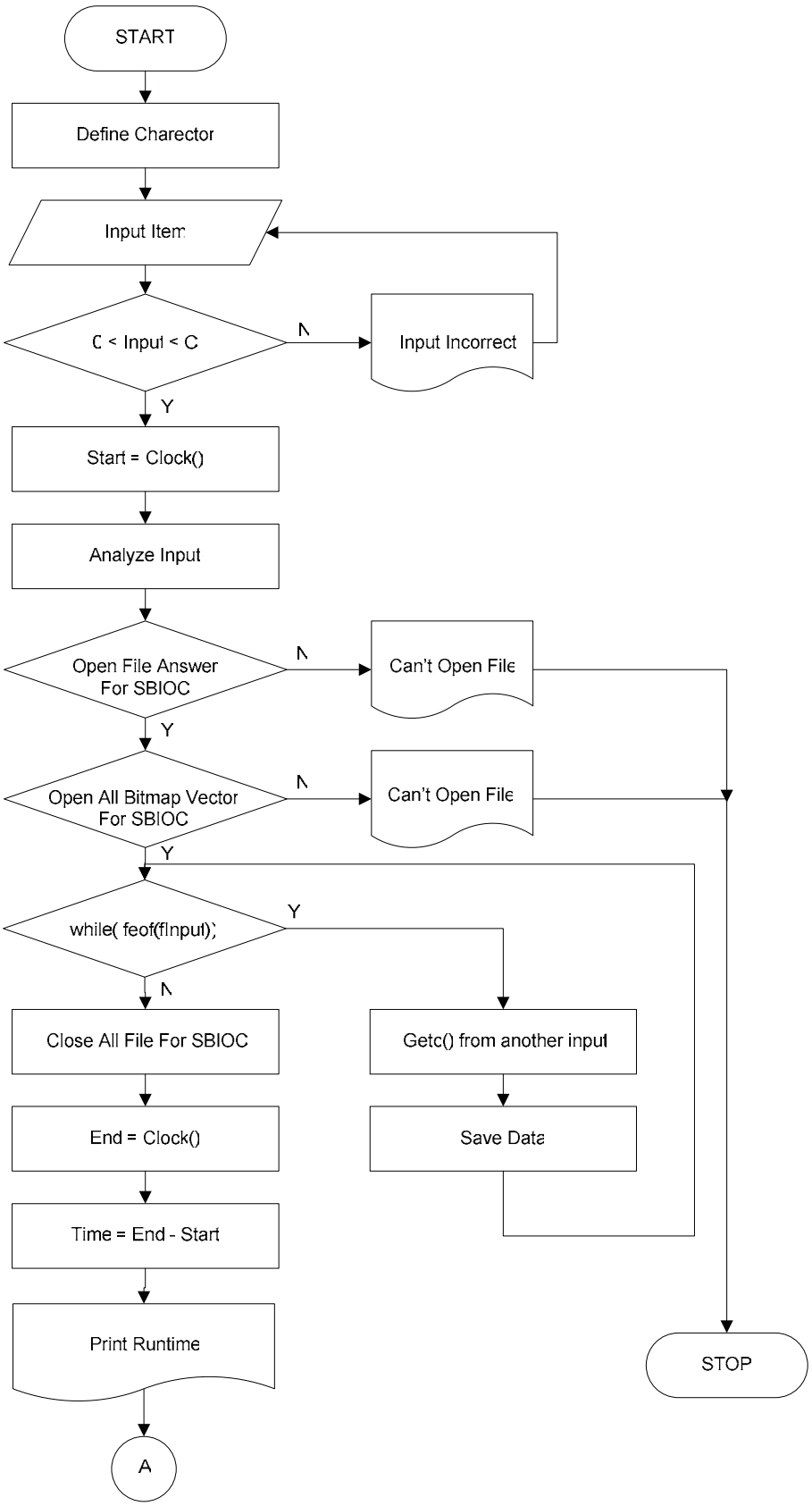
ภาพประกอบ ข-10 วิธีการจับเวลาในการค้นหาข้อมูล

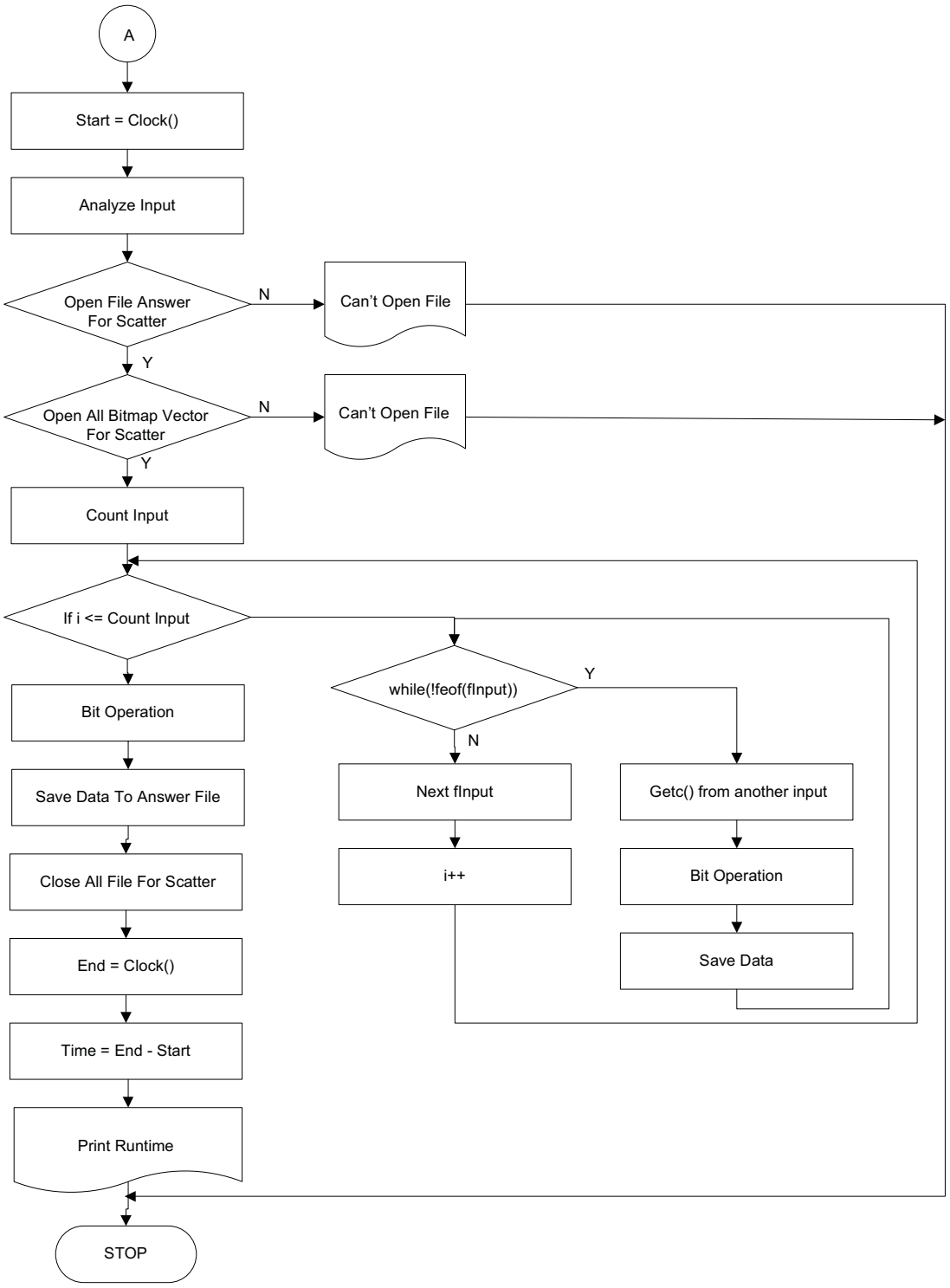
ข.1 แผนผังโปรแกรมการสอบถามกรณีแบบค่าเท่ากัน



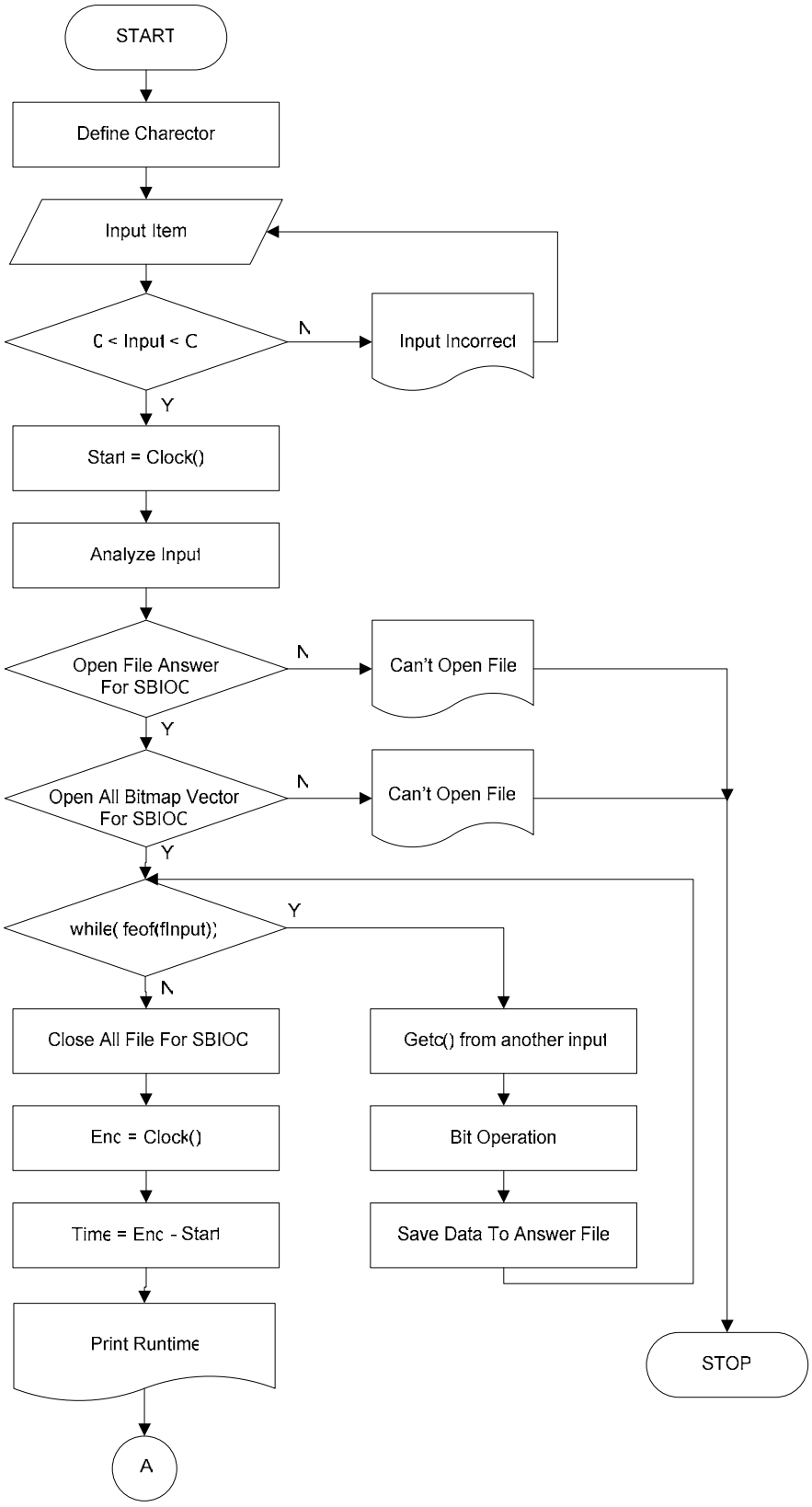

A
Start = Clock()
Analyze Input
Open File Answer For Scatter
N
Can't Open File
Y
Open All Bitmap Vector For Scatter
N
Can't Open File
Y
while(!feof(fInput))
Y
N
Close All File For Scatter
Getc() from another input
End = Clock()
Bit Operation
Time = End - Start
Save Data
Print Runtime
STOP

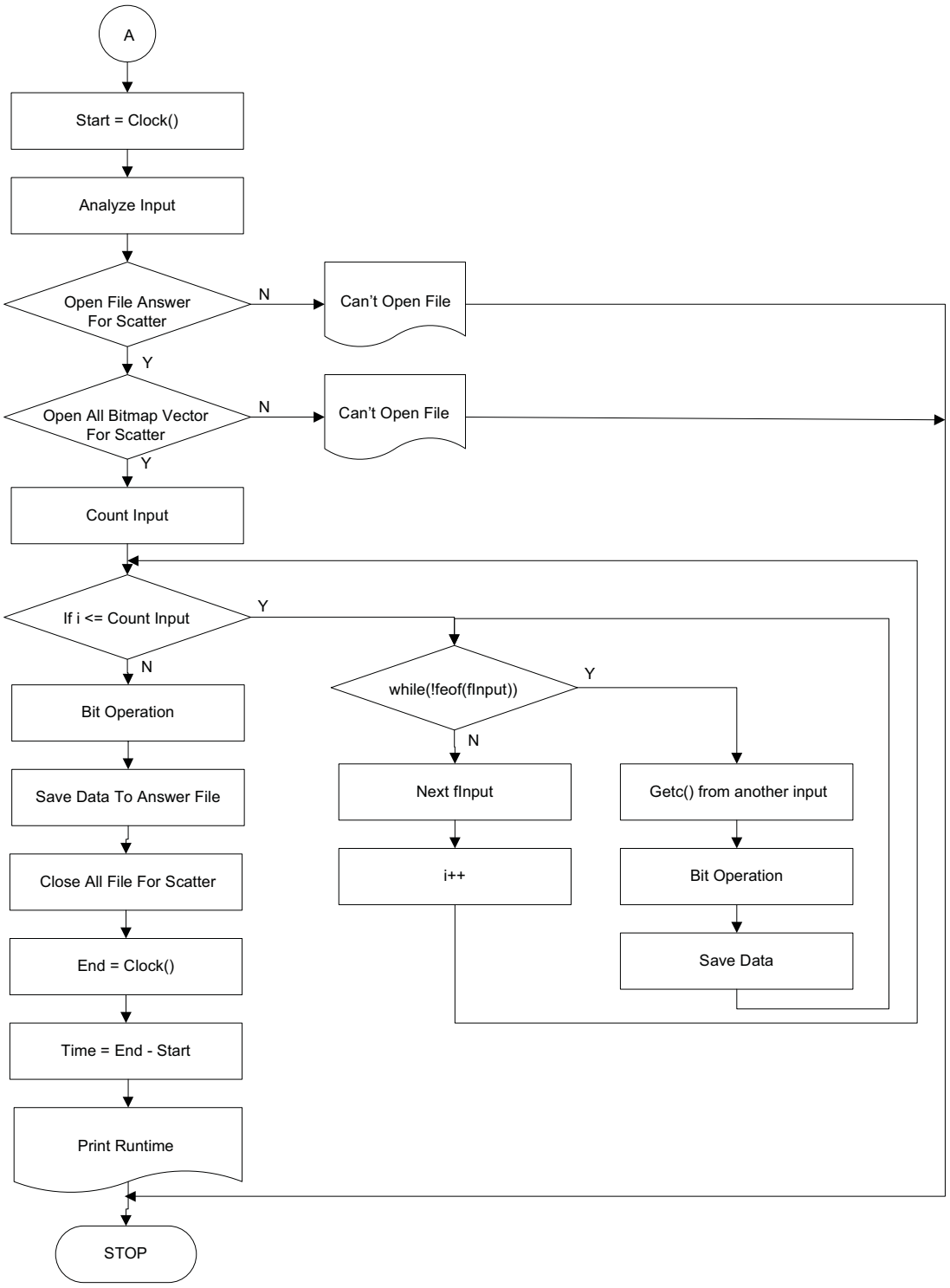
ข.2 แผนผังโปรแกรมการสอบถามกรณีแบบสมาชิกเดียวกัน(search_m.c)



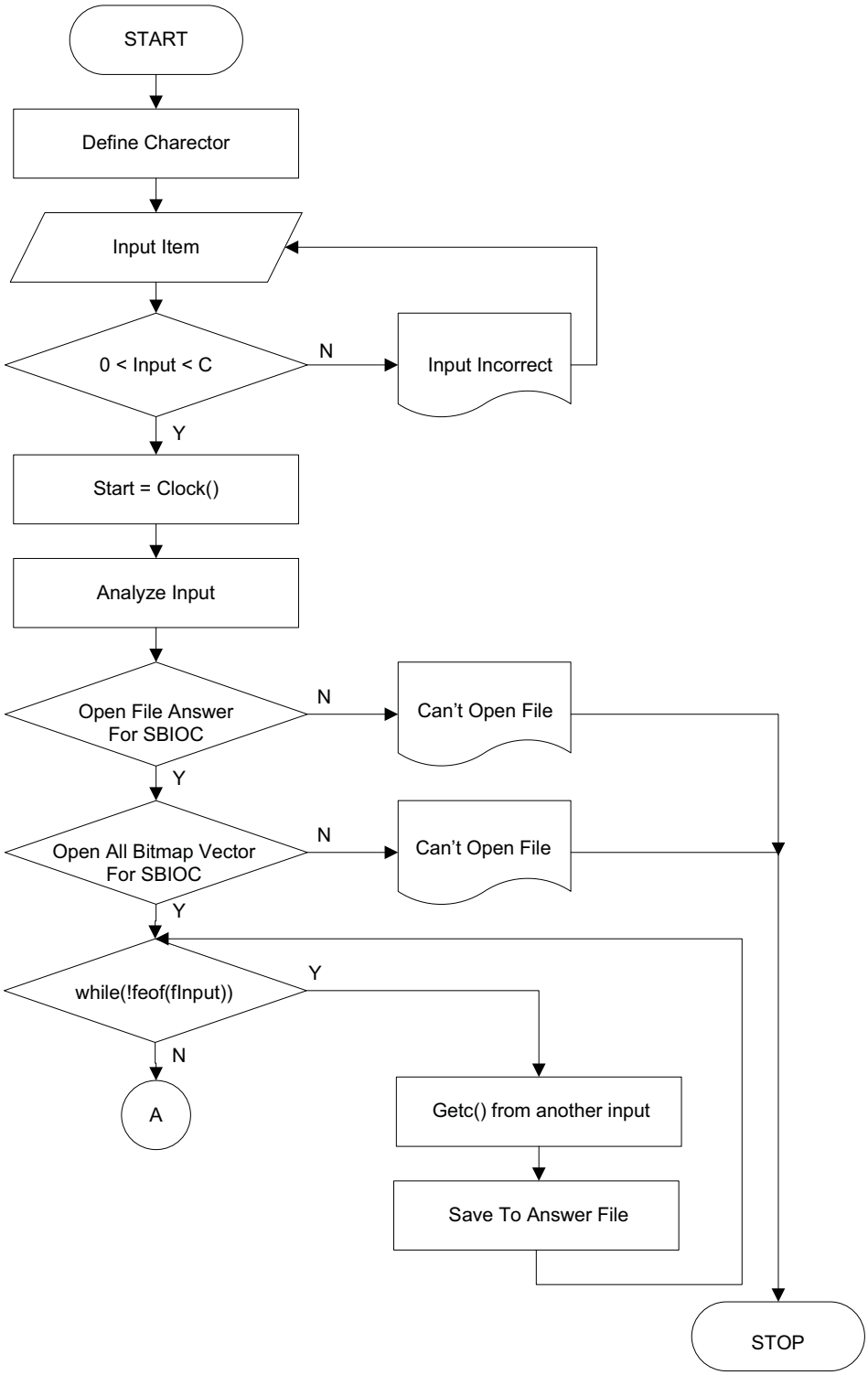

A
Start = Clock()
Analyze Input
Open File Answer For Scatter
N
Can't Open File
Y
Open All Bitmap Vector For Scatter
N
Can't Open File
Y
Count Input
If i <= Count Input
Bit Operation
Save Data To Answer File
Close All File For Scatter
End = Clock()
Time = End - Start
Print Runtime
STOP
while(!feof(fInput))
Y
N
Next fInput
i++
Getc() from another input
Bit Operation
Save Data

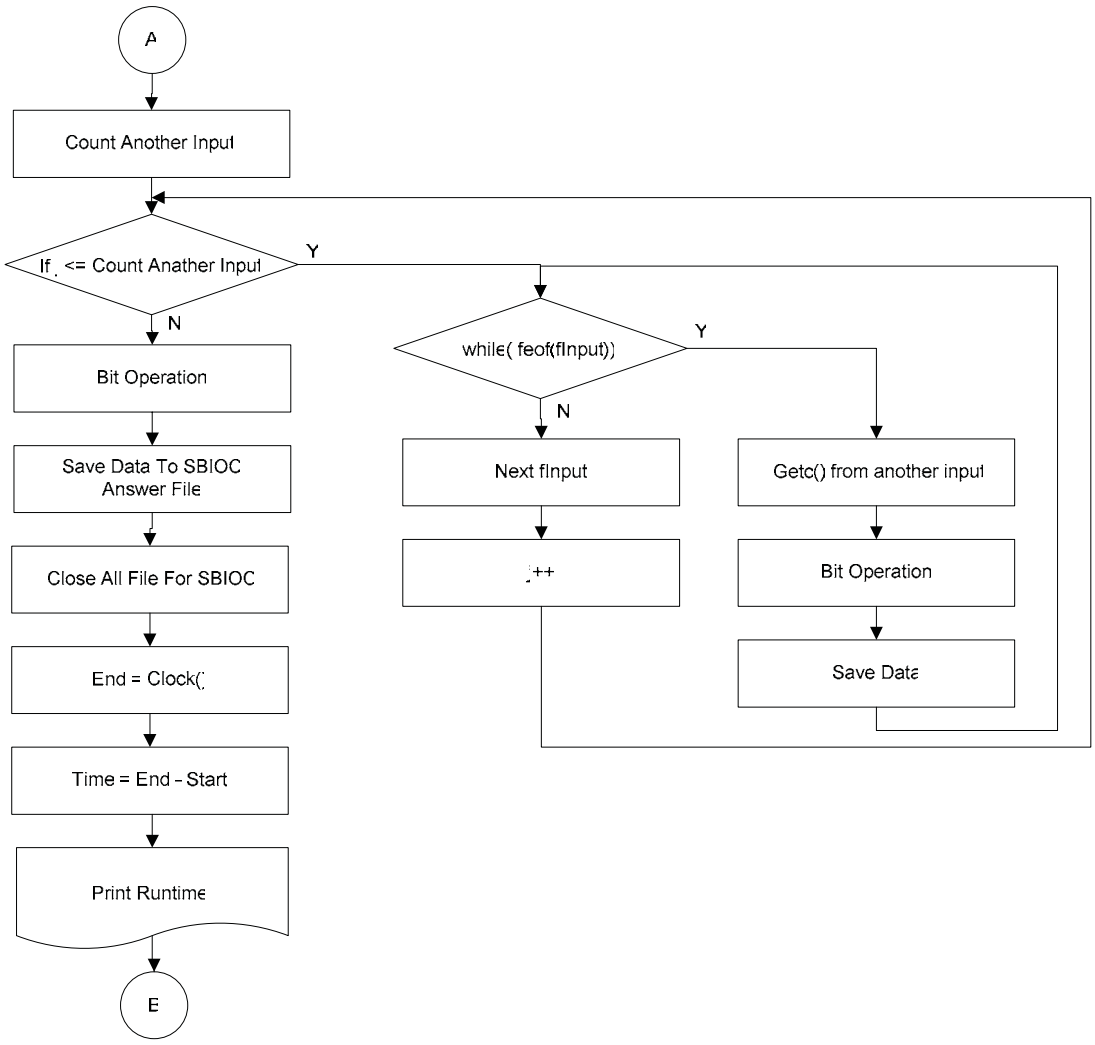
ข.3 แผนผังโปรแกรมการสอบถามแบบสมาชิกเดียวกันกรณีจำนวนสมาชิกเท่ากับจำนวนเท่าของ m(search_2m.c)

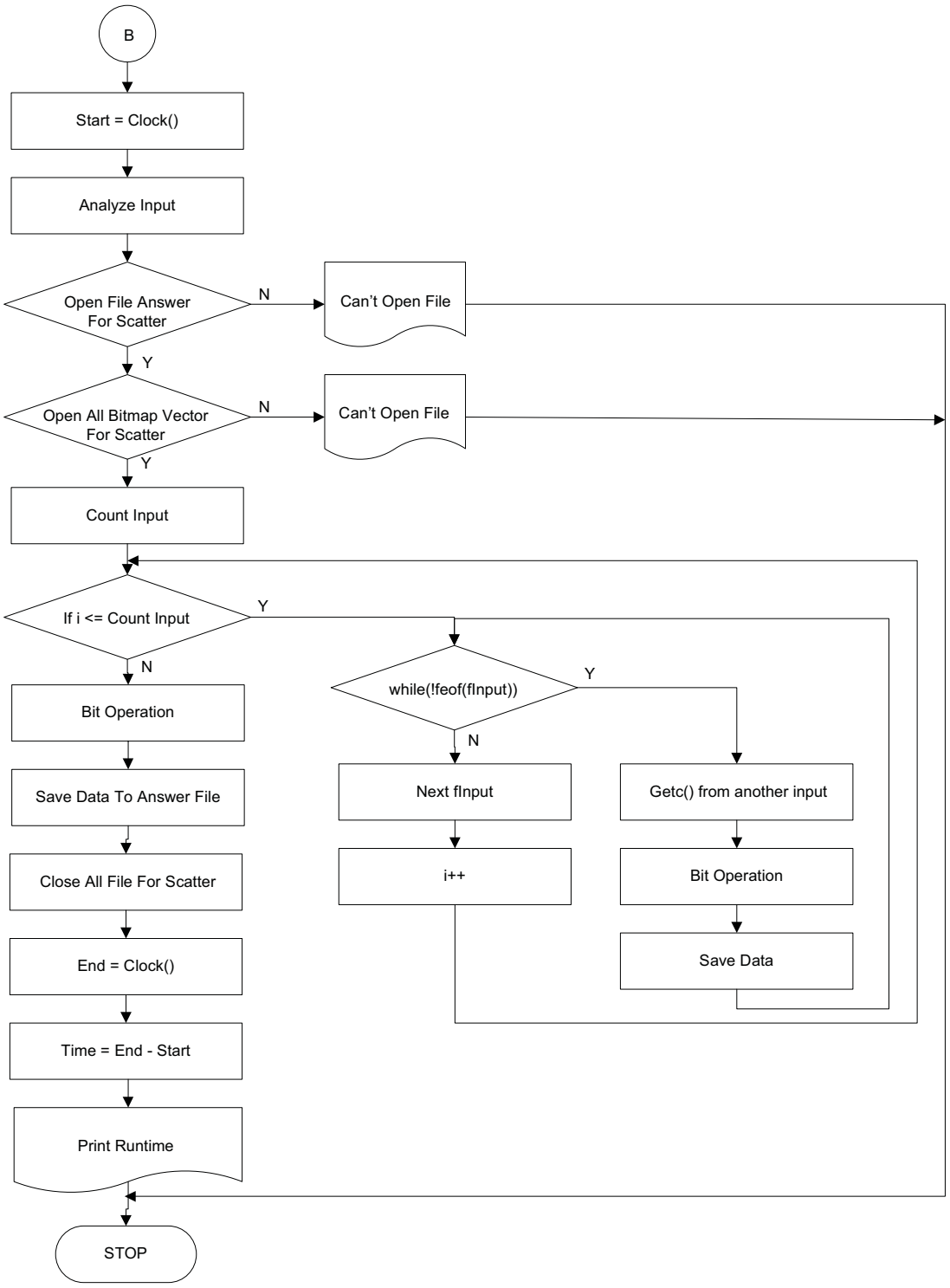



A
Start = Clock()
Analyze Input
Open File Answer For Scatter
N
Can't Open File
Y
Open All Bitmap Vector For Scatter
N
Can't Open File
Y
Count Input
If i <= Count Input
Y
N
Bit Operation
Save Data To Answer File
Close All File For Scatter
End = Clock()
Time = End - Start
Print Runtime
STOP
while(!feof(fInput))
Y
N
Next fInput
i++
Getc() from another input
Bit Operation
Save Data

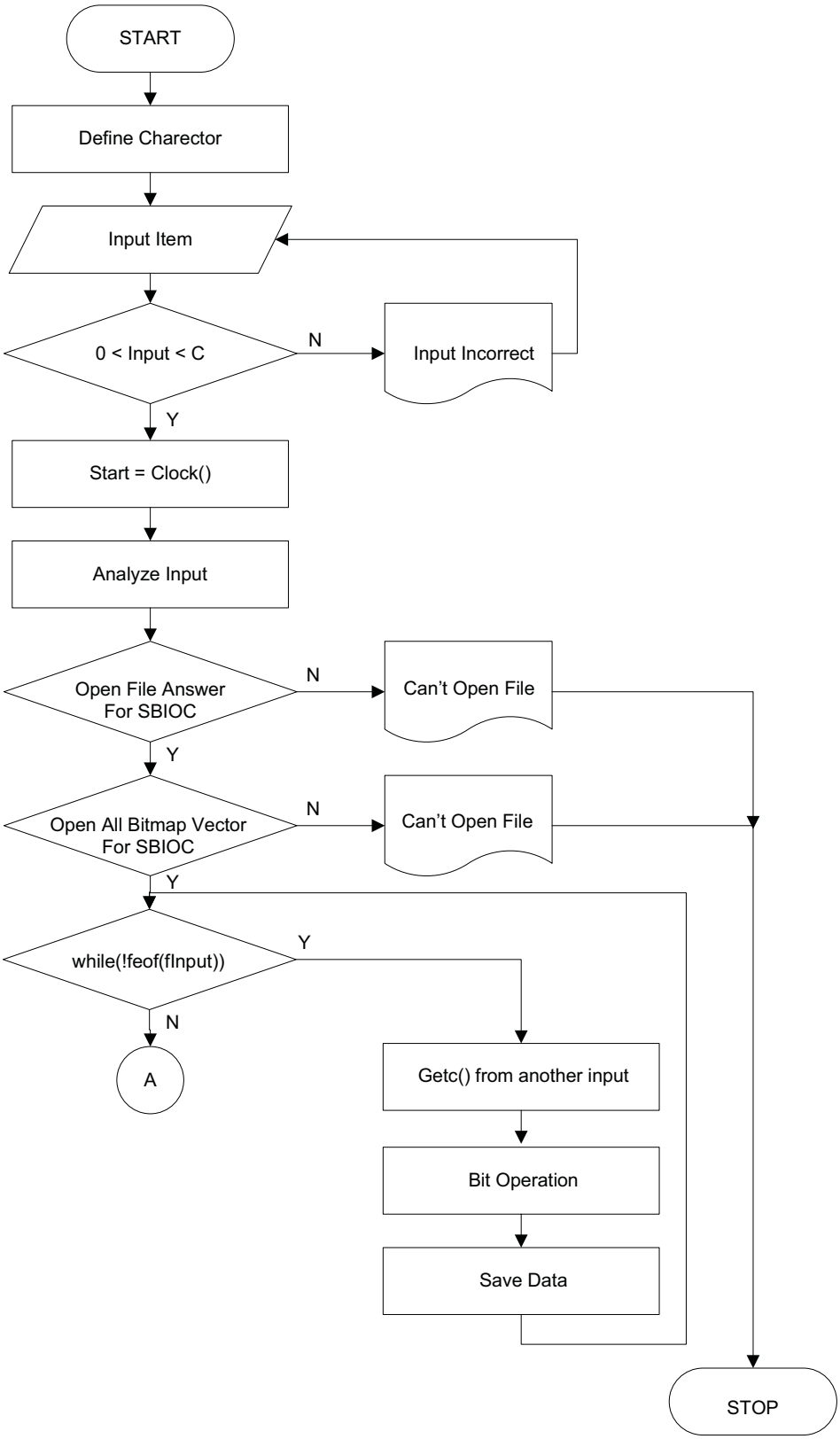
ข.4 แผนผังโปรแกรมการสอบถามแบบสมาชิกเดียวกันกรณีมีบางสมาชิกไม่ได้อยู่กลุ่ม(search_mm1.c)

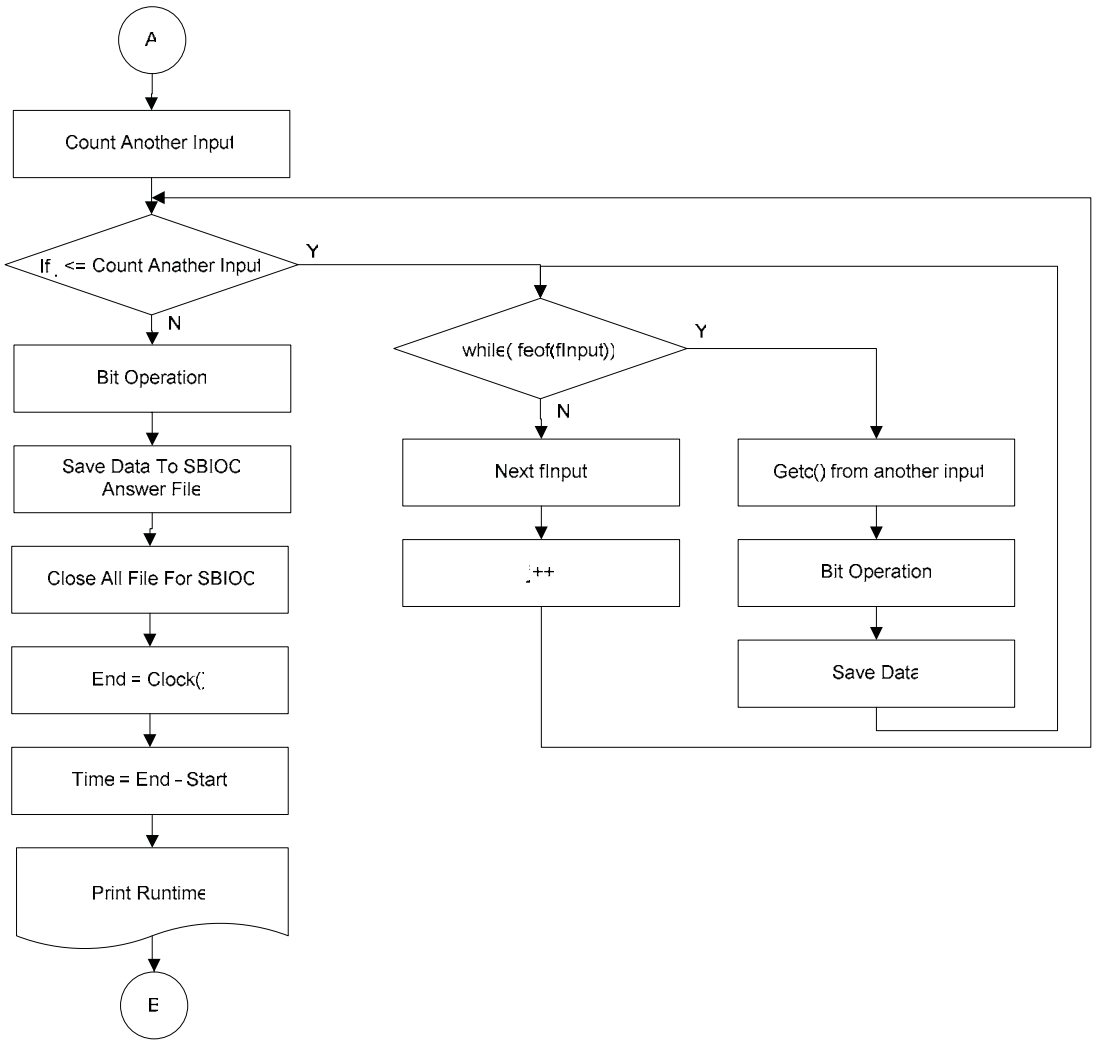


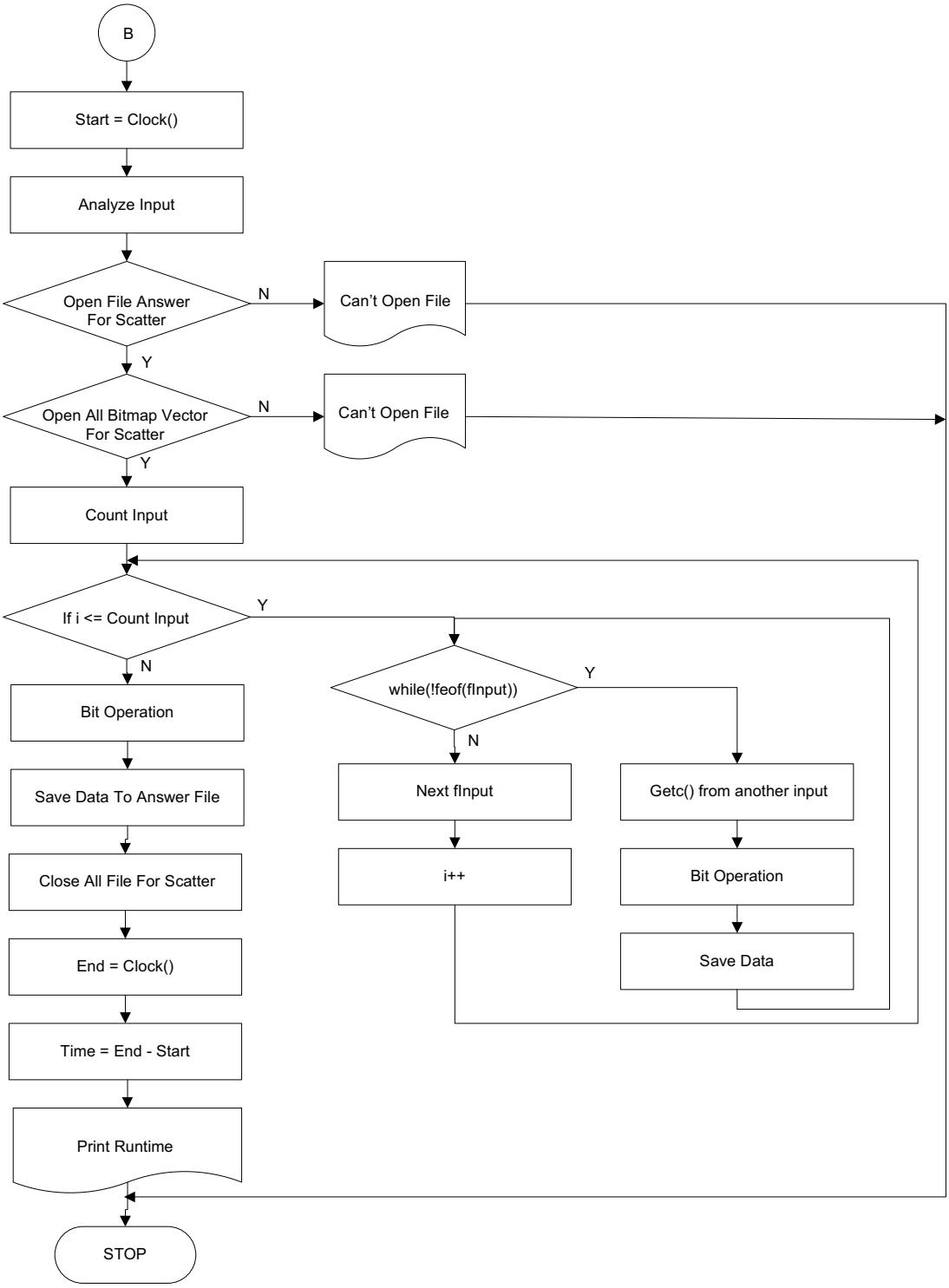

A
Count Another Input
If . <= Count Anather Input
Y
N
Bit Operation
Save Data To SBIOC Answer File
Close All File For SBIOC
End = Clock(
Time = End - Start
Print Runtime
E
while(feof(fInput))
Y
N
Next fInput
.++
Getc() from another input
Bit Operation
Save Data


B
Start = Clock()
Analyze Input
Open File Answer For Scatter
N
Can't Open File
Y
Open All Bitmap Vector For Scatter
N
Can't Open File
Y
Count Input
If i <= Count Input
Y
N
Bit Operation
Save Data To Answer File
Close All File For Scatter
End = Clock()
Time = End - Start
Print Runtime
STOP
while(!feof(fInput))
Y
N
Next fInput
i++
Getc() from another input
Bit Operation
Save Data

ข.5 แผนผังโปรแกรมการสอบถามแบบสมาชิกเดียวกันและมีสมาชิกเท่ากับจำนวนเท่าของ m แต่มีบางสมาชิกไม่ได้อยู่กลุ่ม(search_2mp1.c)




A
Count Another Input
If . <= Count Anather Input
Y
N
Bit Operation
Save Data To SBIOC Answer File
Close All File For SBIOC
End = Clock(
Time = End - Start
Print Runtime
E
while(feof(fInput))
Y
N
Next fInput
.++
Getc() from another input
Bit Operation
Save Data


B
Start = Clock()
Analyze Input
Open File Answer For Scatter
N
Can't Open File
Y
Open All Bitmap Vector For Scatter
N
Can't Open File
Y
Count Input
If i <= Count Input
Y
N
while(!feof(fInput))
Y
N
Next fInput
i++
Getc() from another input
Bit Operation
Save Data
Bit Operation
Save Data To Answer File
Close All File For Scatter
End = Clock()
Time = End - Start
Print Runtime
STOP

ประวัติผู้เขียน

ชื่อ สกุล นายแวฮาซัน แวหะมะ

รหัสประจำตัวนักศึกษา 5010121097

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วศ.บ. (อิเล็กทรอนิกส์)	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2542

ทุนการศึกษาที่ได้รับในระหว่างการศึกษา

ทุนอุดหนุนการศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี ประจำปีการศึกษา 2550-2551

ตำแหน่งและสถานที่ทำงาน

ตำแหน่ง นักวิชาการอุดมศึกษา

สถานที่ทำงาน งานเทคโนโลยีสารสนเทศ คณะศึกษาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี

การตีพิมพ์เผยแพร่ผลงาน

แวฮาซัน แวหะมะ, จรินทร์ หมั่นฝึกพันธ์ และ ศิริรัตน์ วณิชโยบล, "การเพิ่มประสิทธิภาพของดัชนีบิตแมปแบบกระจาย ด้วยเทคนิคการแบ่งกลุ่มข้อมูล", 10 ปีวิชาการแม่ฟ้าหลวง มหาวิทยาลัยแม่ฟ้าหลวง จังหวัดเชียงราย, 26-28 พฤศจิกายน 2551.

W. Weahama, S. Vanichayobon and J. Manfuekphan, "Using Data Clustering to Optimize Scatter Bitmap Index for Membership Queries," International Conference on Computer and Automation Engineering (ICCAE 2009), 2009.