บทที่ 3

ดัชนีแบบบิตแมป(Bitmap Index)

จากบทที่ 2 ที่ได้อธิบายเกี่ยวกับการเข้าถึงข้อมูลโดยใช้ดัชนีไว้ข้างต้น ดัชนีที่นิยมใช้บนระบบฐานข้อมูลดำเนินการ คือ ดัชนีแบบ B-Tree รวมถึงดัชนีที่อยู่บนพื้นฐานของ B-Tree สำหรับบนคลังข้อมูลก็มีใช้เช่นกัน แต่ในกรณีที่แอทริบิวต์มีคาร์ดินอลิตี้ (จำนวนค่าที่เป็นไปได้ของแอทริบิวต์) ต่ำ มักเลือกใช้ดัชนีบิตแมปแทน เช่น แอทริบิวต์เพศ แอทริบิวต์จังหวัด เป็นต้น

ดัชนีบิตแมป(Bitmap Index) เป็นเทคนิควิธีที่ช่วยทำให้เราสามารถค้นหาข้อมูลได้เร็วขึ้น เนื่องจากมีการดำเนินการระดับบิต (AND, OR, XOR, NOT) ระหว่างบิตแมปเวกเตอร์ก่อนดึงข้อมูลจริง ทำให้มีประสิทธิในการประมวลผลเพราะสนับสนุนการทำงานของฮาร์ดแวร์ จึงเหมาะที่จะนำมาใช้กับระบบคลังข้อมูลซึ่งมีข้อมูลปริมาณมากและมักอยู่ในสภาพแวดล้อมที่มีการอ่านสูง ยิ่งกว่านั้นการทำดัชนีบิตแมปยังเป็นการใช้พื้นที่หน่วยความจำได้อย่างมีประสิทธิภาพ โดยเฉพาะอย่างยิ่งกรณีที่แอทริบิวต์ที่มีคาร์ดินอลิตี้ต่ำ [17] สิ่งที่สำคัญในการทำดัชนีบิตแมป คือ รูปแบบการลงรหัส (Encoding Scheme) ซึ่งแต่ละชนิดก็จะมีรูปแบบการลงรหัสที่ต่างกันไป เราพบว่าในงานวิจัยที่ผ่านมามีนำเสนอเทคนิคการทำดัชนีบิตแมปที่น่าสนใจไว้หลายวิธี ซึ่งในบทนี้เราจะกล่าวถึง 4 ชนิด ได้แก่

- ดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index)
- ดัชนีบิตแมปแบบช่วง (Interval Bitmap Index)
- ดัชนีบิตแมปแบบกระจาย (Scatter Bitmap Index)
- ดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index)

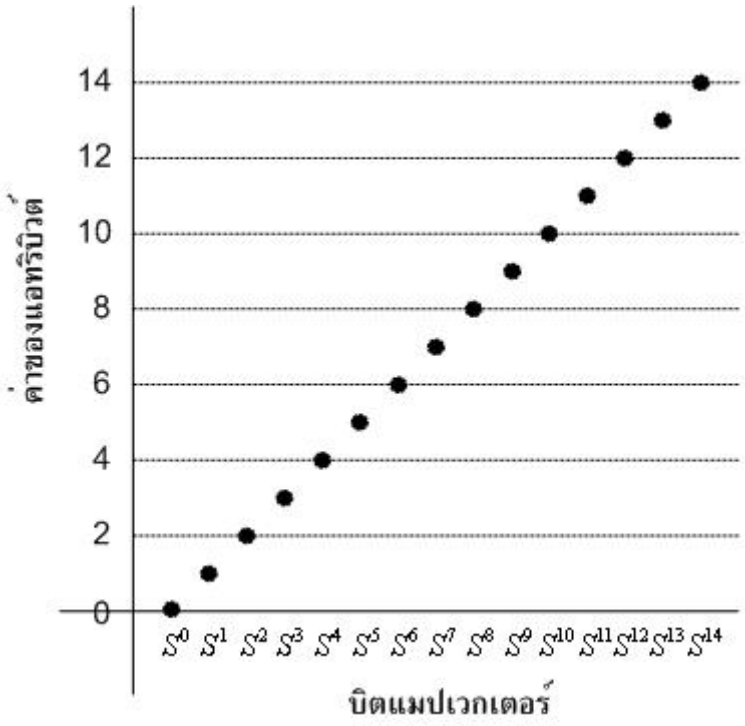
เพื่อความสะดวกในการพิจารณาต่อไป จึงมีการกำหนดเงื่อนไขและสัญลักษณ์ดังต่อไปนี้

1. กำหนดให้ C แทน จำนวนค่าที่เป็นไปได้ของแอทริบิวต์ที่เลือกมาทำดัชนีบิตแมป

2. กำหนดให้ v แทน ให้ค่าที่เป็นไปได้ของแอทริบิวต์ที่เลือกมาทำดัชนีบิตแมป ซึ่งจะต้องเป็นจำนวนเต็มที่มีความต่อเนื่องกันโดยเริ่มตั้งแต่ 0 ดังนั้น ค่าที่เป็นไปได้ของแอทริบิวต์ที่นำทำดัชนี คือ $v \in \{0,1,2,...,C-1\}$

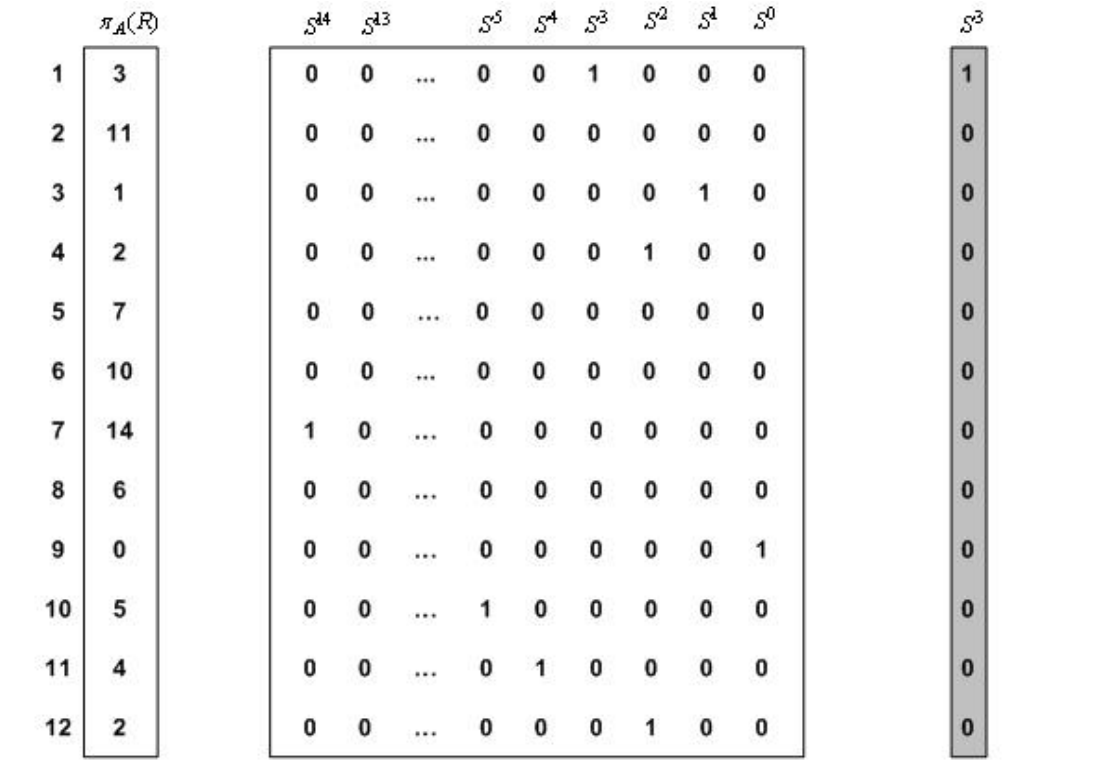
3.1 ดัชนีบิตแมปแบบพื้นฐาน (Simple Bitmap Index)

ดัชนีบิตแมปแบบพื้นฐานมีการแทนค่าข้อมูลโดยใช้ 1 บิตแมปเวกเตอร์ต่อ 1 ค่าของแอทริบิวต์ ดังนั้น การทำดัชนีบิตแมปแบบพื้นฐานจึงประกอบด้วย C บิตแมปเวกเตอร์ คือ บิตแมปเวกเตอร์ $S^0, S^1, S^2, ..., S^{C-1}$ โดยที่ S^v แทนบิตแมปเวกเตอร์ที่มีค่าของแอทริบิวต์เท่ากับ v ซึ่งในการแทนค่าจะกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ S^v มีค่าเป็น 1 ก็ต่อเมื่อเรคอร์ดที่ i มีค่า v ส่วนบิตอื่นที่เหลือมีค่าเป็น 0 [15,17,16] พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบพื้นฐานดังภาพประกอบ 3-1



ภาพประกอบ 3-1 แสดงแผนภาพการลงรหัสดัชนีบิตแมปแบบพื้นฐาน เมื่อ $C = 15$ ใช้ 15 บิตแมปเวกเตอร์

จากภาพประกอบ 3-1 แอทริบิวต์ A มีค่าที่เป็นไปได้ 15 ค่า คือ 0,1,2,...,14 ดังนั้นดัชนีบิตแมปแบบพื้นฐานประกอบด้วย 15 บิตแมปเวกเตอร์ คือ $S^0, S^1, S^2, ..., S^{14}$ โดยที่แต่ละบิตแมปเวกเตอร์แทนแต่ละค่าของแอทริบิวต์ A เช่น บิตแมปเวกเตอร์ S^3 แทนค่าของแอทริบิวต์ A ที่มีค่าเท่ากับ 3 พิจารณาตัวอย่างดัชนีบิตแมปแบบพื้นฐานดังภาพประกอบ 3-2



	$\pi_A(R)$
1	3
2	11
3	1
4	2
5	7
6	10
7	14
8	6
9	0
10	5
11	4
12	2

(ก) แอทริบิวต์ A

S^{14}	S^{13}	...	S^5	S^4	S^3	S^2	S^1	S^0
0	0	...	0	0	1	0	0	0
0	0	...	0	0	0	0	0	0
0	0	...	0	0	0	0	1	0
0	0	...	0	0	0	1	0	0
0	0	...	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0
1	0	...	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0
0	0	...	0	0	0	0	0	1
0	0	...	1	0	0	0	0	0
0	0	...	0	1	0	0	0	0
0	0	...	0	0	0	1	0	0

(ข) ดัชนีบิตแมปแบบพื้นฐานบนแอทริบิวต์ A

S^3
1
0
0
0
0
0
0
0
0
0
0
0

(ค) การสอบถาม "$A = 3$"

ภาพประกอบ 3-2 แสดงดัชนีบิตแมปแบบพื้นฐานบนแอทริบิวต์ A เมื่อ $C = 15$ ใช้ 15 บิตแมปเวกเตอร์

จากภาพประกอบ 3-2 เรคอร์ดที่ 1 ของแอทริบิวต์ A มีค่าเท่ากับ 3 ดังนั้น บิตที่ 1 บิตแมปเวกเตอร์ S^3 จึงมีค่าเท่ากับ 1 ส่วนบิตอื่นที่เหลือของบิตแมปเวกเตอร์ v มีค่าเท่ากับ 0

3.1.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบพื้นฐาน

การสอบถามแบบค่าเท่ากันมีรูปแบบ คือ "$A = v$" หมายถึง การสอบถามว่าบนแอทริบิวต์ A มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ v สำหรับการสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบพื้นฐาน สามารถทำได้โดยการอ่านค่าบิตแมปเวกเตอร์ที่สัมพันธ์กับค่า v ซึ่งก็คือบิตแมปเวกเตอร์ S^v นั่นเอง ดังนั้นการสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบพื้นฐานมีรูปแบบทั่วไปคือ

$$"A = v" = S^v$$

เมื่ออ่านค่าบิตแมปเวกเตอร์ S^v แล้วพบว่าบิตใดมีค่าเท่ากับ 1 แสดงค่าเรคอร์ดนั้นคือคำตอบ

ตัวอย่างการสอบถามพิจารณาจากภาพประกอบ 3-2 ต้องการทราบว่าบนแอทริบิวต์ A มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 3 จะได้ว่า $"A=3"=S^3$ ดังนั้นจึงต้องอ่านบิตแมปเวกเตอร์ S^3 พบว่า บิตที่ 1 มีค่าเท่ากับ 1 ส่วนบิตอื่น ๆ มีค่าเท่ากับ 0 จึงไว้ว่า เรคอร์ดที่มีค่าของแอทริบิวต์เป็น 3 คือเรคอร์ดที่ 1

3.1.2 ข้อดีของดัชนีบิตแมปแบบพื้นฐาน

เนื่องจาก B-tree เหมาะสำหรับระบบฐานข้อมูล ซึ่งต้องมีกลไกในการค้นหา นอกจากนี้ยังใช้เวลามากและพื้นที่ในการจัดเก็บดัชนีเป็นแบบลอกการิทึม แต่ B-tree สามารถทำงานได้ดีในระบบที่มักมีการเปลี่ยนแปลงข้อมูลอยู่เสมอ และนั่นไม่ใช่คุณสมบัติที่ต้องการสำหรับระบบคลังข้อมูล ดังนั้นเมื่อพิจารณาในระบบคลังข้อมูลแล้ว สำหรับแอทริบิวต์ที่มีคาร์ดินอลิตี้ต่ำ จะได้ว่าการทำดัชนีบิตแมปแบบพื้นฐานดีกว่าการทำดัชนีแบบ B-tree เพราะการสร้างดัชนีบิตแมปแบบพื้นฐานว่าใช้เวลาและพื้นที่ในน้อยกว่า และการทำงานของดัชนีบิตแมปแบบพื้นฐานช่วยลดเวลาในการค้นหาก่อนที่จะเข้าถึงข้อมูลจริง

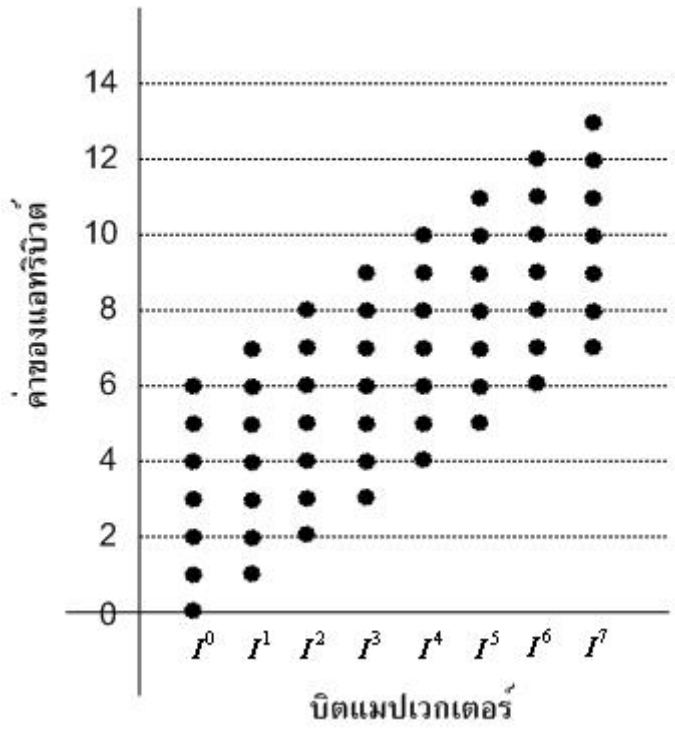
เมื่อพิจารณาถึงการสอบถาม เราพบว่าการทำดัชนีทั้ง 2 แบบ ต่างก็ช่วยให้การประมวลผลรวดเร็วยิ่งขึ้น แต่เมื่อมีการสอบถามที่มีการกำหนดเงื่อนไขที่มากกว่า 1 เงื่อนไข เช่น $A=v_1$ AND $B=v_2$ การทำงานของดัชนีแบบ B-tree จะไม่มีประสิทธิภาพ เพราะจะต้องมีการสร้างคีย์ผสม (Compound Key) แต่ในทางกลับกัน ดัชนีบิตแมปแบบพื้นฐานจะใช้การดำเนินการทางตรรกะ AND ระหว่างบิตแมปเวกเตอร์ ซึ่งมีความรวดเร็วกว่ามาก

3.1.3 ข้อจำกัดของดัชนีบิตแมปแบบพื้นฐาน

อย่างไรก็ตามเมื่อคาร์ดินอลิตี้ของแอทริบิวต์เพิ่มสูงขึ้น จะทำให้พื้นที่ที่ใช้ในการจัดเก็บดัชนีบิตแมปแบบพื้นฐานมากขึ้นและใช้ประโยชน์ของพื้นที่ได้ไม่เต็มที่ ซึ่งปัญหาที่แอทริบิวต์มีคาร์ดินอลิตี้สูงขึ้นนี้เอง ได้นำไปสู่การคิดค้นเทคนิคการทำดัชนีบิตแมปแบบอื่น ๆ ตามมา

3.2 ดัชนีบิตแมปแบบช่วง (Interval Bitmap Index)

จากข้อจำกัดของการทำดัชนีบิตแมปแบบพื้นฐานนี้เอง จึงนำไปสู่การคิดค้นดัชนีบิตแมปแบบช่วงที่ใช้พื้นที่ในการจัดเก็บดัชนีน้อยกว่าเดิม ซึ่งการทำดัชนีบิตแมปแบบช่วงประกอบด้วย $\lceil C/2 \rceil$ บิตแมปเวกเตอร์ คือ $I^0, I^1, ..., I^{\lceil C/2 \rceil - 1}$ โดยที่ I^j แทนบิตแมปเวกเตอร์ที่มีค่าของแอทริบิวต์อยู่ในช่วง $[j, j+m]$ (เขียนแทนด้วย $I^j = [j, j+m]$) และ $m = \lfloor C/2 \rfloor - 1$ ซึ่งในการแทนค่าจะกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ I^j มีค่าเป็น 1 ก็ต่อเมื่อเรคอร์ดที่ i มีค่าของแอทริบิวต์อยู่ในช่วง $[j, j+m]$ ส่วนบิตอื่นที่เหลือมีค่าเป็น 0 [17] พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบช่วงดังภาพประกอบ 3-3

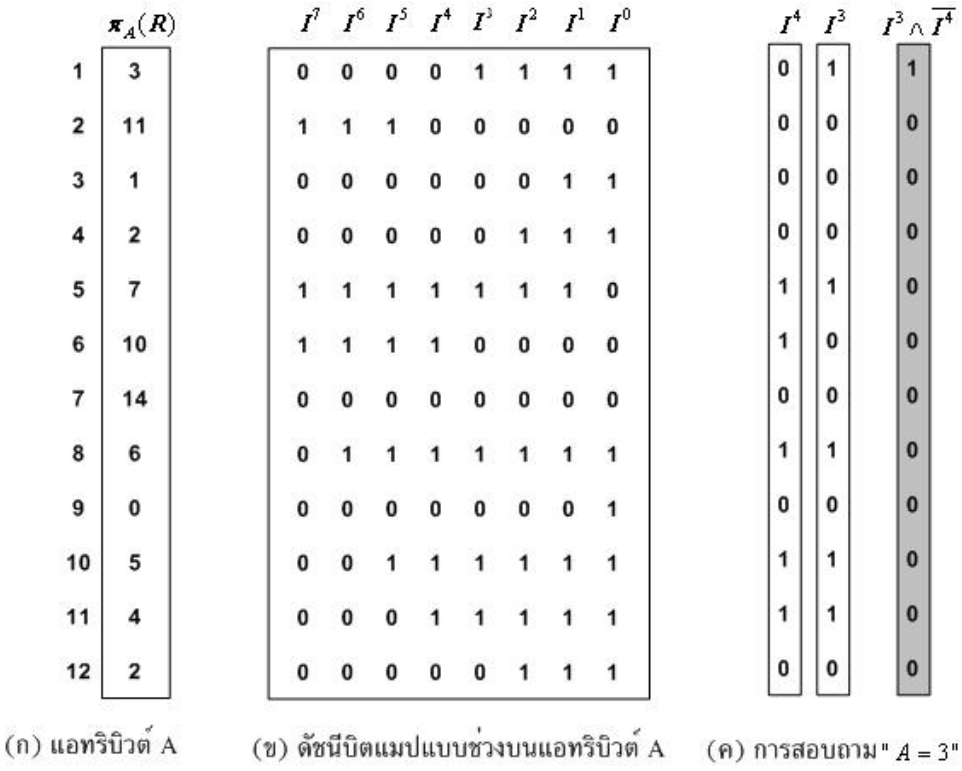


ภาพประกอบ 3-3 แสดงแผนภาพการลงรหัสดัชนีบิตแมปแบบช่วง เมื่อ $C=15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-3 แอทริบิวต์ A 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 A ที่อยู่ในช่วง $[j, j+6]$ หรือ $I^j = [j, j+6]$ จึงได้ว่า

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

พิจารณาตัวอย่างดัชนีบิตแมปแบบช่วงดังภาพประกอบ 3-4



	$\pi_A(R)$
1	3
2	11
3	1
4	2
5	7
6	10
7	14
8	6
9	0
10	5
11	4
12	2

(ก) แอทริบิวต์ A

	I^7	I^6	I^5	I^4	I^3	I^2	I^1	I^0
1	0	0	0	0	1	1	1	1
2	1	1	1	0	0	0	0	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	1	1
5	1	1	1	1	1	1	1	0
6	1	1	1	1	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	0	0	0	0	0	0	1
10	0	0	1	1	1	1	1	1
11	0	0	0	1	1	1	1	1
12	0	0	0	0	0	1	1	1

(ข) ดัชนีบิตแมปแบบช่วงบนแอทริบิวต์ A

	I^4	I^3	$I^3 \wedge \overline{I^4}$
	0	1	1
	0	0	0
	0	0	0
	0	0	0
	1	1	0
	1	0	0
	0	0	0
	1	1	0
	0	0	0
	1	1	0
	1	1	0
	0	0	0

(ค) การสอบถาม "$A = 3$"

ภาพประกอบ 3-4 แสดงดัชนีบิตแมปแบบช่วงบนแอทริบิวต์ A เมื่อ $C = 15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-4 เรคอร์ดที่ 1 ของแอทริบิวต์ A 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

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

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

3.2.2 ข้อดีของดัชนีบิตแมปแบบช่วง

การทำดัชนีบิตแมปแบบช่วงใช้พื้นที่ในการจัดเก็บดัชนีน้อยกว่าการทำดัชนีบิตแมปแบบพื้นฐานมาก กล่าวคือสามารถลดพื้นที่ลงได้ประมาณครึ่งหนึ่ง

3.2.3 ข้อจำกัดของดัชนีบิตแมปแบบช่วง

ประสิทธิภาพในการสอบถามแบบค่าเท่ากับของดัชนีบิตแมปแบบช่วงยังด้อยกว่าดัชนีบิตแมปแบบพื้นฐาน กล่าวคือ การสอบถามแบบค่าเท่ากับของดัชนีบิตแมปแบบช่วงส่วนใหญ่ต้องใช้การดำเนินการตรรกะระหว่าง 2 บิตแมปเวกเตอร์(อ่านเพิ่มขึ้น 1 บิตแมปเวกเตอร์) แต่การสอบถามแบบค่าเท่ากันของดัชนีบิตแมปแบบพื้นฐานใช้การตรวจสอบเพียง 1 บิตแมปเวกเตอร์เท่านั้น นอกจากนั้น สำหรับการทำดัชนีบิตแมปเพื่อใช้ในการสอบถามแบบค่าเท่ากัน จะเห็นว่าจากแผนภาพการลงรหัสดัชนีบิตแมปแบบช่วงในภาพประกอบ 3-3 ยังสามารถลดพื้นที่ลงได้อีกเนื่องจากยังมีส่วนของบิตแมปเวกเตอร์ที่ใช้แทนค่าข้อมูลของแอทริบิวต์ซ้ำกันมากเกินไป

3.3 ดัชนีบิตแมปแบบกระจาย (Scatter Bitmap Index)

การทำดัชนีบิตแมปแบบกระจายเป็นเทคนิคใหม่ที่เหมาะสำหรับแอทริบิวต์ที่มักมีการสอบถามแบบค่าเดียว ซึ่งการทำดัชนีบิตแมปแบบกระจาย ประกอบด้วย $\lceil 2\sqrt{C} \rceil$ บิตแมปเวกเตอร์ แบ่งเป็น 2 กลุ่ม คือ กลุ่ม Z มี $\left\lfloor \frac{C}{m-1} \right\rfloor + 1$ บิตแมปเวกเตอร์ และกลุ่ม L มี $m-2$ บิตแมปเวกเตอร์ โดยที่ $m = \lceil \sqrt{C} \rceil + 1$

แต่ละค่าของแอทริบิวต์จะถูกแทนด้วยบิตแมปเวกเตอร์ที่อยู่ในกลุ่ม Z และกลุ่ม L หรืออาจถูกแทนด้วยบิตแมปเวกเตอร์ที่อยู่ในกลุ่ม Z อย่างเดียว ซึ่งในการทำดัชนีมีวิธีการดังนี้

<u>การสร้างบิตแมปเวกเตอร์กลุ่ม Z</u>

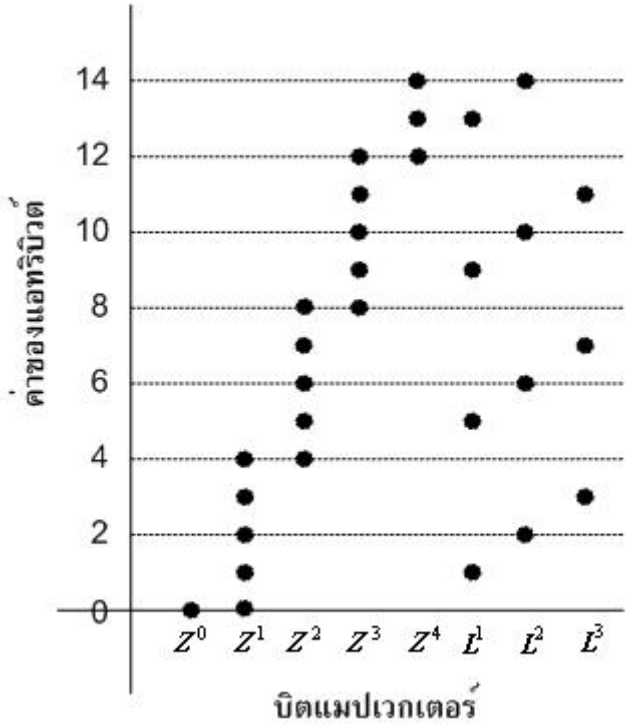
- สร้างบิตแมปเวกเตอร์ Z^0 โดยกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ Z^0 มีค่าเป็น 1 ก็ต่อเมื่อ เรคอร์ดที่ i มีค่าของแอทริบิวต์เท่ากับ 0 ส่วนบิตอื่นที่เหลือมีค่าเป็น 0
- สร้างบิตแมปเวกเตอร์ Z^j โดยกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ Z^j มีค่าเป็น 1 ก็ต่อเมื่อ $j=\left\lfloor \frac{v}{m-1} \right\rfloor +1$ โดยที่ เรคอร์ดที่ i มีค่าของแอทริบิวต์เท่ากับ v ส่วนบิตอื่นที่เหลือของบิตแมปเวกเตอร์ Z^j มีค่าเป็น 0

และถ้า $m-1$ หาร v ลงตัว แล้วจะกำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ Z^{j-1} มีค่าเป็น 1 ด้วย

<u>การสร้างบิตแมปเวกเตอร์กลุ่ม L</u>

- กำหนดให้บิตที่ i ของบิตแมปเวกเตอร์ L^k มีค่าเป็น 1 ก็ต่อเมื่อ $k=v \bmod(m-1)$ โดยที่เรคอร์ดที่ i มีค่าของแอทริบิวต์เท่ากับ v และ $k \neq 0$ [19, 20]

พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบกระจายดังภาพประกอบ 3-5



ภาพประกอบ 3-5 แสดงแผนภาพการลงรหัสดัชนีบิตแมปแบบกระจาย เมื่อ $C=15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-5 แอทริบิวต์ A มีค่าที่เป็นไปได้ 15 ค่า คือ 0,1,2,...,14 ดังนั้นดัชนีบิตแมปแบบกระจายประกอบด้วย $8\left(=\left\lceil 2\sqrt{15}\right\rceil\right)$ 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 3-6

	$\pi_A(R)$
1	3
2	11
3	1
4	2
5	7
6	10
7	14
8	6
9	0
10	5
11	4
12	2

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

L^3	Z^1	$Z^1 \wedge L^3$
1	1	1
1	0	0
0	1	0
0	1	0
1	0	0
0	0	0
0	0	0
0	0	0
0	1	0
0	0	0
0	1	0
0	1	0

(ก) แอทริบิวต์ A (ข) ดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ A (ค) การสอบถาม "$A = 3$"

ภาพประกอบ 3-6 แสดงดัชนีบิตแมปแบบกระจายบนแอทริบิวต์ A เมื่อ $C = 15$ ใช้ 8 บิตแมปเวกเตอร์

จากภาพประกอบ 3-6 เรคอร์ดที่ 1 ของแอทริบิวต์ A 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

3.3.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบกระจาย

การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบกระจายมีรูปแบบทั่วไปคือ

$$"A = v" = \begin{cases} Z^{\overline{\frac{v}{m-1}}} \wedge Z^{\overline{\frac{v}{(m-1)+1}}} & \text{ถ้า } v \bmod (m-1) = 0, \\ Z^{\overline{\frac{v}{(m-1)}}+1} \wedge L^{v \bmod (m-1)} & \text{กรณีอื่น ๆ} \end{cases}$$

ตัวอย่างเช่น จากภาพประกอบ 3-6 ต้องการทราบว่า บนแอทริบิวต์ A มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 3 ซึ่งตรงกับเงื่อนไขที่ $2\,(3 \bmod (5-1) \neq 0)$ ดังนั้น คำตอบสามารถหาได้จาก

$$"A = 3" = Z^{\overline{\frac{v}{(m-1)}}+1} \wedge L^{v \bmod (m-1)}$$

$$= Z^{\overline{\frac{3}{(5-1)}}+1} \wedge L^{3 \bmod (5-1)}$$

$$= Z^1 \wedge L^3$$

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 1

3.3.1 ข้อดีของดัชนีบิตแมปแบบกระจาย

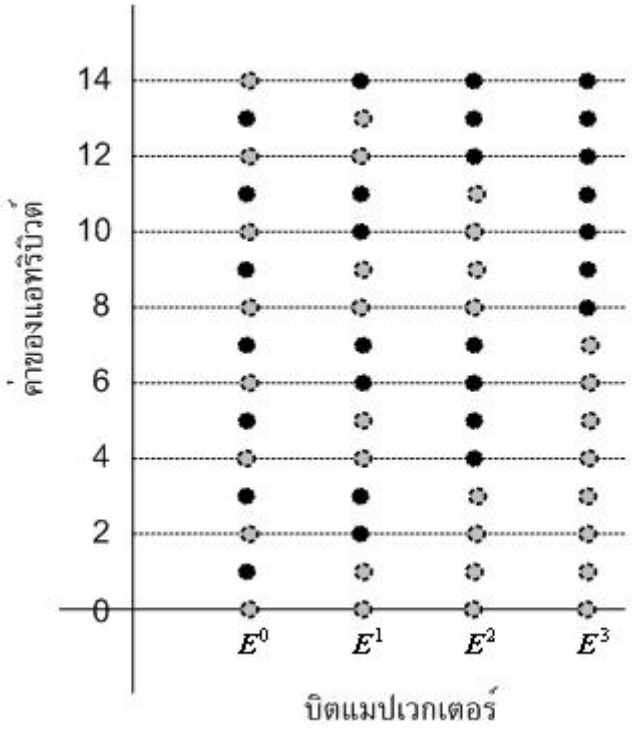
การทำดัชนีบิตแมปแบบกระจายใช้พื้นที่น้อยกว่าดัชนีบิตแมปแบบช่วง และความสามารถในการสอบถามแบบค่าเท่ากันของดัชนีบิตแมปแบบกระจายก็ไม่ด้อยไปกว่าดัชนีบิตแมปแบบช่วง คือยังใช้การดำเนินการตรรกะระหว่าง 2 บิตแมปเวกเตอร์

3.3.2 ข้อจำกัดของดัชนีบิตแมปแบบกระจาย

จากแผนภาพการทำดัชนีบิตแมปแบบกระจายในภาพประกอบ 3-5 จะเห็นได้ว่าสามารถลดพื้นที่ได้อีก เนื่องจากยังมีบางบิตแมปเวกเตอร์ที่ใช้แทนค่าข้อมูลของแอทริบิวต์ได้น้อยเกินไป โดยเฉพาะบิตแมปเวกเตอร์ Z^0

3.4 ดัชนีบิตแมปแบบเข้ารหัส (Encoded Bitmap Index)

การทำดัชนีบิตแมปแบบเข้ารหัส มีวิธีการสร้างดัชนีและการสอบถามที่แตกต่างไปจากวิธีที่ได้กล่าวไว้ข้างต้นโดยสิ้นเชิง ซึ่งการทำดัชนีบิตแมปแบบเข้ารหัสประกอบด้วย $\lceil \log_2 C \rceil$ บิตแมปเวกเตอร์ คือ $E^0, E^1, ..., E^{\lceil \log_2 C \rceil}$ โดยการลงรหัสข้อมูลใช้วิธีเทียบค่าจากตารางการเทียบค่า(Mapping Table) เพื่อกำหนดค่าให้กับแต่ละบิตแมปเวกเตอร์ บิตที่ i ของแต่ละบิตแมปเวกเตอร์ จะมีค่าเท่ากับ 0 หรือ 1 ขึ้นอยู่กับค่าของแอทริบิวต์เรคอร์ดที่ i ว่า เทียบค่าของแต่ละบิตแมปเวกเตอร์ตรงกับค่าใด นั่นหมายความว่า แต่ละค่าของแอทริบิวจะถูกแทนค่าด้วยทุกบิตแมปเวกเตอร์[16] พิจารณาแผนภาพการลงรหัสดัชนีบิตแมปแบบเข้ารหัสดังภาพประกอบ 3-7



ภาพประกอบ 3-7 แสดงแผนภาพการลงรหัสดัชนีบิตแมปแบบเข้ารหัส เมื่อ $C = 15$ ใช้ 4 บิตแมปเวกเตอร์

จากภาพประกอบ 3-7 แอทริบิวต์ A มีค่าที่เป็นไปได้ 15 ค่า คือ 0,1,2,...,14 ดังนั้นดัชนีบิตแมปแบบกระจายประกอบด้วย $4\left(=\lceil \log_2 15 \rceil\right)$ บิตแมปเวกเตอร์ คือ E^0, E^1, E^2, E^3 แต่ละค่าของแอทริบิวต์ถูกแทนด้วย 4 บิตแมปเวกเตอร์ พิจารณาตัวอย่างดัชนีบิตแมปแบบเข้ารหัสดังภาพประกอบ 3-8

	$E^3E^2E^1E^0$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0

(ก) ตารางการเทียบค่า

	$\pi_A(R)$
1	3
2	11
3	1
4	2
5	7
6	10
7	14
8	6
9	0
10	5
11	4
12	2

(ข) แอทริบิวต์ A

E^3	E^2	E^1	E^0
0	0	1	1
1	0	1	1
0	0	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	0
0	1	1	0
1	1	0	0
0	0	0	0
0	1	0	0
0	0	1	0

(ค) ดัชนีบิตแมปแบบเข้ารหัสบนแอทริบิวต์ A

ภาพประกอบ 3-8 แสดงดัชนีบิตแมปแบบเข้ารหัสบนแอทริบิวต์ A เมื่อ $C = 15$ ใช้ 4 บิตแมปเวกเตอร์

จากภาพประกอบ 3-8 เรคอร์ดที่ 1 ของแอทริบิวต์ A มีค่าเท่ากับ 3 ซึ่งเมื่อเทียบค่ากับตารางการเทียบค่าได้ตรงกับรูปแบบ $E^3E^2E^1E^0 = 0\ 0\ 1\ 1$ ดังนั้น บิตที่ 1 ของบิตแมปเวกเตอร์ E^3 และ E^2 มีค่าเท่ากับ 1 ส่วนบิตที่ 1 ของบิตแมปเวกเตอร์ E^1 และ E^0 มีค่าเท่ากับ 0

3.4.1 การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบเข้ารหัส

ในการสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปแบบเข้ารหัส สามารถทำได้โดยการนำค่าที่ต้องการสอบถามไปเทียบค่ากับตารางการเทียบค่าว่าแต่ละบิตแมปเวกเตอร์เก็บค่าใด จากนั้นนำมาเทียบกับทุก ๆ เรคอร์ดบนตารางดัชนี ถ้าเรคอร์ดใดมีค่าที่ตรงกันกับค่าได้จากตารางการเทียบทุกบิตแมปเวกเตอร์ แล้วเรคอร์ดนั้นคือคำตอบ

จะเห็นได้ว่า การสอบถามแบบค่าเท่ากันบนดัชนีบิตแมปต้องมีการอ่านข้อมูลและดำเนินการตรรกเป็นจำนวนมาก ตัวอย่างเช่น จากภาพประกอบ 3-8 ต้องการทราบว่า บนแอทริบิวต์ A มีเรคอร์ดใดบ้างที่มีค่าเท่ากับ 3 สามารถทำได้โดยการเทียบค่า 3 กับตารางการเทียบค่าได้ตรงกับรูปแบบ 0 0 1 1 ($E^3E^2E^1E^0$) จากนั้นตรวจสอบทุกเรคอร์ดบนตารางดัชนีว่าเรคคอร์ดใด้บ้างมีรูปแบบเป็น 0 0 1 1 ($E^3E^2E^1E^0$) ผลลัพธ์ที่ได้คือ เรคอร์ดที่ 1 เป็นคำตอบ

3.4.2 ข้อดีของดัชนีบิตแมปแบบเข้ารหัส

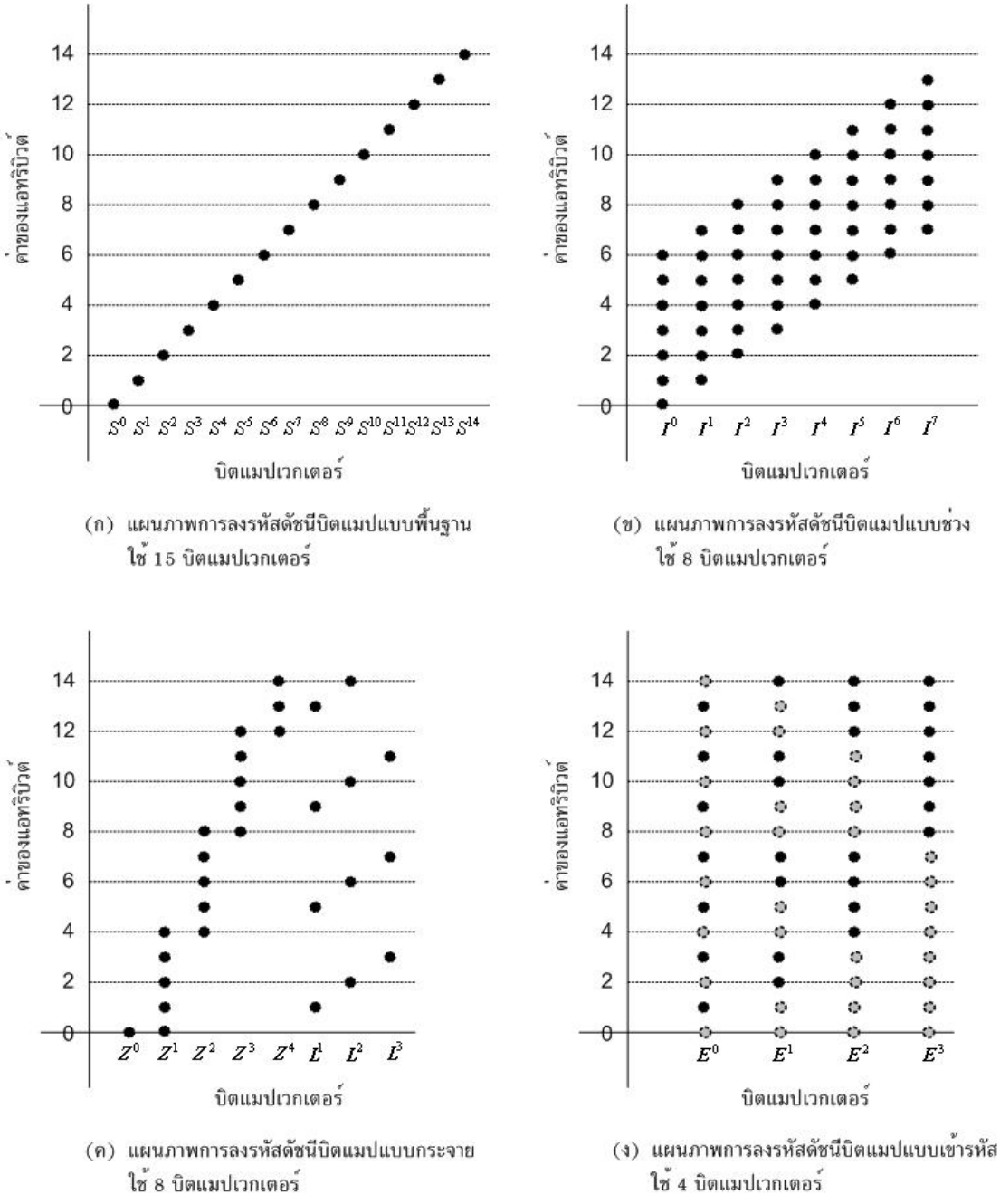
การทำดัชนีบิตแมปแบบเข้ารหัสมุ่งเน้นไปที่การใช้พื้นที่ให้มีประสิทธิภาพสูงสุด เมื่อพิจารณาการทำดัชนีแบบบิตแมปทุกวิธีที่ได้กล่าวมาข้างต้นแล้ว จะเห็นได้ว่าการทำดัชนีบิตแมปแบบเข้ารหัสใช้พื้นที่น้อยที่สุด ซึ่งส่งผลให้เราสามารถเก็บข้อมูลได้มากขึ้นหรือดำเนินการทำดัชนีบิตแมปกับแอทริบิวต์ที่มีคาร์ดินอลิตี้ที่สูงได้ นอกจากนั้นยังทำให้สามารถรับและส่งข้อมูลได้ครั้งละมาก ๆ จึงเสียค่าใช้จ่ายในการรับและส่งข้อมูลน้อยลง

3.4.3 ข้อจำกัดของดัชนีบิตแมปแบบเข้ารหัส

แม้ว่าการทำดัชนีบิตแมปแบบเข้ารหัสจะสามารถลดพื้นที่ในการจัดเก็บดัชนีบิตแมปได้มาก แต่ในการสอบถามนั้นต้องมีการเปรียบเทียบกับทุกแถวของข้อมูลว่าตรงกันหรือไม่ จึงไม่สามารถใช้การดำเนินการทางตรรกะระหว่างบิตแมปเวกเตอร์ที่อยู่บนดัชนีได้ ต้องมีการอ่านทุกบิตแมปเวกเตอร์ จึงทำให้ต้องใช้เวลานานมาก

3.5 เปรียบเทียบดัชนีบิตแมปแบบทั้ง 4 ชนิด

จากที่ได้กล่าวมา ดัชนีบิตแมปแต่ละชนิดมีการลงรหัสและสอบถามที่แตกต่างกันไป จึงมีข้อดีและข้อจำกัดที่แตกต่างกันไป พิจารณารูปแบบการลงรหัสที่แตกต่างกันของดัชนีบิตแมปแต่ละชนิดดังภาพประกอบ 3-9



ภาพประกอบ 3-9 แสดงแผนภาพการลงรหัสดัชนีบิตแมปแบบทั้ง 4 ชนิด เมื่อ $C = 15$

จากภาพประกอบ 3-9 การทำดัชนีบิตแมปแบบพื้นฐานใช้บิตแมปเวกเตอร์มากที่สุด คือ เท่ากับจำนวนคาร์ดินอลิตี้ ในที่นี้คาร์ดินอลิตี้เท่ากับ 15 ดัชนีบิตแมปแบบพื้นฐานจึงใช้ 15 บิตแมปเวกเตอร์ดังภาพประกอบ 3-9(ก) ส่วนดัชนีบิตแมปแบบเข้ารหัสใช้บิตแมปเวกเตอร์น้อยที่สุด คือ 4 บิตแมปเวกเตอร์ดังภาพประกอบ 3-9(ง) สำหรับดัชนีบิตแมปแบบช่วงและดัชนีบิตแมปแบบกระจายใช้ 8 บิตแมปเวกเตอร์เท่ากัน ซึ่งในกรณีที่คาร์ดินอลิตีเท่ากับ 15 นี้จะ

ยังใช้เท่ากันอยู่ แต่เมื่อคาร์ดินอลิตี้มีค่ามากขึ้น ดัชนีบิตแมปแบบกระจายจะใช้บิตแมปเวกเตอร์น้อยกว่า ($\lceil 2\sqrt{C} \rceil < \lceil C/2 \rceil$ เมื่อ $C \geq 19$) พิจารณาการเปรียบเทียบลักษณะที่สำคัญของดัชนีบิตแมปแต่ละชนิดดังตาราง 3-1

ตาราง 3-1 แสดงลักษณะที่สำคัญของดัชนีบิตแมปแบบพื้นฐาน แบบช่วง แบบกระจาย และแบบเข้ารหัส (C คือ คาร์ดินอลิตี้ของแอทริบิวต์ที่เลือกมาทำดัชนี)

ลักษณะที่สำคัญ	ดัชนีบิตแมป			
	แบบพื้นฐาน	แบบช่วง	แบบกระจาย	แบบเข้ารหัส
จำนวนบิตแมปเวกเตอร์ที่ใช้ในการแทนแต่ละค่าของแอทริบิวต์	1	$\leq \lfloor C/2 \rfloor$	2	$\lceil \log_2 C \rceil$
จำนวนค่าของแอทริบิวต์ที่ถูกแทนด้วยแต่ละบิตแมปเวกเตอร์	1	$\lfloor C/2 \rfloor$	$\leq \lceil \sqrt{C} \rceil + 1$	C
จำนวนบิตแมปเวกเตอร์ที่ใช้ในการแทนค่าของแอทริบิวต์ที่มีคาร์ดินอลิตี้ C (พื้นที่)	C	$\lceil C/2 \rceil$	$\lceil 2\sqrt{C} \rceil$	$\lceil \log_2 C \rceil$
จำนวนบิตแมปเวกเตอร์ที่อ่าน : จำนวนครั้งในการดำเนินการตรรกะสำหรับการสอบถามแบบค่าเท่ากัน (เวลา)	1:0	2:2 (1AND,1NOT)	2:1 (1AND)	$\lceil \log_2 C \rceil$: ใช้การเทียบค่าจากตารางการเทียบค่า

จากรูปแบบการลงรหัสค่าของแอทริบิวต์และลักษณะที่สำคัญของดัชนีบิตแมปแต่ละชนิด ในภาพประกอบ 3-9 และตาราง 3-1 สามารถเปรียบเทียบประสิทธิภาพในเรื่องการใช้พื้นที่ในการจัดเก็บดัชนีและเวลาที่ใช้ในการสอบถามแบบค่าเท่ากันได้ดังนี้

- การใช้พื้นที่ในการจัดเก็บดัชนี พิจารณาจากจำนวนบิตแมปเวกเตอร์ทั้งหมดที่ใช้ในการแทนค่าของแอทริบิวต์
  - ดัชนีบิตแมปแบบเข้ารหัสใช้พื้นที่น้อยที่สุด
  - ดัชนีบิตแมปแบบพื้นฐานใช้พื้นที่มากที่สุด

- ดัชนีบิตแมปแบบช่วงใช้พื้นที่เป็นครึ่งหนึ่งของดัชนีบิตแมปแบบพื้นฐาน
- เมื่อคาร์ดินอลิตี้สูงขึ้น ($C \geq 19$) ดัชนีบิตแมปแบบกระจายใช้พื้นที่น้อยกว่าดัชนีบิตแมปแบบช่วง

- เวลาที่ใช้ในการสอบถามแบบค่าเท่ากัน พิจารณาจากจำนวนบิตแมปเวกเตอร์ที่อ่านและจำนวนครั้งในการดำเนินการตรรกะ
  - ดัชนีบิตแมปแบบพื้นฐานใช้เวลาน้อยที่สุด
  - ดัชนีบิตแมปแบบเข้ารหัสใช้เวลามากที่สุด
  - ดัชนีบิตแมปแบบช่วงและดัชนีบิตแมปกระจายอ่านเพียง 2 บิตแมปเวกเตอร์เท่านั้น โดยดัชนีบิตแมปแบบกระจายใช้การดำเนินการตรรกะ 1 ครั้ง แต่ในบางเงื่อนไขของการสอบถามบนดัชนีบิตแมปแบบช่วงต้องมีการดำเนินการตรรกะถึง 2 ครั้ง

จากทั้งหมดที่ได้กล่าวมา จะเห็นได้ว่า ดัชนีบิตแมปแบบพื้นฐานใช้เวลาในการสอบถามน้อยที่สุด แต่กลับต้องสูญเสียพื้นที่ในการจัดเก็บดัชนีมากที่สุด ส่วนดัชนีบิตแมปแบบเข้ารหัสประหยัดพื้นที่ในการจัดเก็บดัชนีมากที่สุด แต่เวลาในการสอบถามช้าที่สุด อีกทางเลือกหนึ่งในการทำดัชนี คือ ดัชนีบิตแมปแบบช่วงและแบบกระจาย ซึ่งสามารถประหยัดพื้นที่ได้มาก แต่ยังคงมีประสิทธิภาพในการสอบถามอยู่ เนื่องจากใช้เพียงการดำเนินการทางตรรกะในการสอบถามเท่านั้น และสำหรับการสอบถามแบบค่าเท่ากัน จะได้ว่า ดัชนีบิตแมปแบบกระจายมีความคุ้มค่ามากกว่าดัชนีบิตแมปแบบช่วง เพราะใช้พื้นที่น้อยกว่า นอกจากนั้นในบางเงื่อนไขก็ใช้การดำเนินการตรรกะน้อยกว่าด้วย

เมื่อพิจารณาภาพประกอบ 3-9 แล้วจะเห็นว่า รูปแบบการลงรหัสดัชนีบิตแมปแบบช่วงมีการใช้หลายบิตแมปเวกเตอร์ในการแต่ละค่าของแอทริบิวต์ ซึ่งการลงรหัสของดัชนีบิตแมปแบบกระจาย มีการปรับให้มีลดความสิ้นเปลืองในการใช้บิตแมปเวกเตอร์แทนแต่ละค่าของแอทริบิวต์ โดยใช้เพียง 2 บิตแมปเวกเตอร์เท่านั้น จึงสามารถลดพื้นที่ในการจัดเก็บดัชนีได้อีกเมื่อแอทริบิวต์มีคาร์ดินอลิตี้ที่สูงขึ้น อย่างไรก็ตาม ดัชนีบิตแมปแบบกระจายก็ยังการใช้ประโยชน์ของแต่ละบิตแมปเวกเตอร์ได้ไม่คุ้มค่า กล่าวคือ แต่ละบิตแมปเวกเตอร์ใช้แทนค่าของแอทริบิวต์ได้น้อยค่า โดยเฉพาะอย่างยิ่ง บิตแมปเวกเตอร์ Z^0 ใช้แทนค่า 0 เพียงค่าเดียวเท่านั้น ผู้วิจัยได้คิดค้นเทคนิคการทำดัชนีบิตแมปแบบใหม่ขึ้นที่สามารถลดพื้นที่ในการจัดเก็บดัชนีได้อีก ซึ่งดัชนีที่สร้างขึ้นใหม่นี้มีชื่อว่า ดัชนีบิตแมปแบบคู่กัน (Dual Bitmap Index)