

การเก็บข้อมูลของระบบฐานข้อมูลแบบกระจาย

1. Distributed Data Storage

ในการจัดเก็บข้อมูลในระบบฐานข้อมูลแบบกระจายมีอยู่หลายวิธีด้วยกันคือ

1.1 การทำสำเนา(Replication) เป็นการทำสำเนาของรีเลชันไว้หลาย ๆ สำเนา และแต่ละสำเนาจะถูกเก็บไว้ต่างไซต์กัน

1.2 การแยกรีเลชัน(Fragmentation) เป็นการแยกรีเลชันออกเป็นหลาย ๆ ส่วน และจัดเก็บแต่ละส่วนไว้ต่างไซต์กัน

1.3 วิธี Replication และ Fragmentation เป็นการแยกรีเลชันออกเป็นหลาย ๆ ส่วน และแต่ละส่วนก็จะมีจัดการแบบสำเนา

1.1 การทำสำเนาข้อมูล(Data Replication)

วิธีการเก็บข้อมูลวิธีนี้เป็นการทำสำเนาของรีเลชัน r ไว้หลาย ๆ ไซต์ โดยทั่วไปจะทำการสำเนาแบบ Full Replication นั่นคือจะทำการสร้างสำเนาของรีเลชันกับทุก ๆ ไซต์ในระบบในการจัดเก็บข้อมูลแบบสำเนา มีทั้งข้อดีและข้อเสียดังนี้คือ

- Availability ถ้าไซต์ที่มีรีเลชัน r เกิดหยุดการทำงานลง ระบบสามารถที่จะทำงานต่อไปได้โดยไปดึงข้อมูล รีเลชัน r ที่ไซต์อื่น โดยไม่ต้องคำนึงถึงไซต์ที่หยุดทำงานไป

- Increased parallelism ในกรณีที่การดำเนินการกับข้อมูลส่วนใหญ่ของรีเลชัน r เป็นการอ่านข้อมูลจากรีเลชัน เมื่อเรามีข้อมูลของรีเลชัน r อยู่ในหลาย ๆ ไซต์ เราก็สามารถที่จะทำการดึงข้อมูลมาพร้อม กันได้ อีกทั้งจากการที่มีสำเนาของรีเลชันอยู่หลาย ๆ ไซต์ โอกาสที่จะพบข้อมูลในไซต์ที่ต้องการจะทำทรานแซกชันก็มีโอกาสมากกว่า ดังนั้นด้วยวิธีการนี้จะช่วยลดปริมาณข้อมูลที่จะส่งผ่านระหว่างไซต์ได้

- Increased overhead on update ด้วยวิธีการ replication นี้ สำเนาข้อมูลของรีเลชัน r ในแต่ละไซต์จะต้องเหมือนกันทุก ๆ ไซต์ ไม่เช่นนั้นแล้วการประมวลผลกับรีเลชัน r อาจจะทำให้เกิด

ความผิดพลาดขึ้นได้ ดังนั้นเมื่อไรก็ตามที่รีเลชัน r ถูกแก้ไข ระบบจะต้องทำการแก้ไขข้อมูลรีเลชัน r ให้ครบทุกไซต์ที่มีรีเลชัน r อยู่ ยกตัวอย่างเช่น ในระบบธนาคารซึ่งมีการทำสำเนาของข้อมูลบัญชีไว้หลาย ๆ ไซต์ เมื่อมีการปรับปรุงยอดบัญชี ก็จะต้องทำการปรับยอดบัญชีให้เท่ากันในทุก ๆ ไซต์

โดยทั่วไปแล้วการทำสำเนาข้อมูลจะช่วยเพิ่มประสิทธิภาพในการอ่านข้อมูล และช่วยเพิ่มความสะดวกในการสืบค้นข้อมูลของทรานแซกชันแบบอ่านอย่างเดียว อย่างไรก็ตามทรานแซกชันแบบที่มีการปรับปรุงข้อมูลจะต้องมีการดำเนินการที่เพิ่มมากขึ้น ซึ่งการควบคุมการปรับปรุงข้อมูลพร้อม ๆ กันในหลาย ๆ ไซต์ค่อนข้างจะยุ่งยากกว่าระบบฐานข้อมูลแบบรวมศูนย์

1.2 การแยกข้อมูล(Data Fragmentation)

วิธีการเก็บข้อมูลแบบนี้ รีเลชัน r จะถูกแบ่งออกเป็นรีเลชันย่อย ๆ r_1, r_2, \dots, r_n ซึ่งรีเลชันเหล่านี้เมื่อนำกลับมารวมกันจะได้ผลลัพธ์เป็นรีเลชัน r เหมือนเดิม ซึ่งในการแบ่งรีเลชันมีอยู่ 2 วิธี คือ การแบ่งรีเลชันตามแนวนอน(horizontal fragmentation) และ การแบ่งรีเลชันตามแนวตั้ง(vertical fragmentation) โดยการแบ่งรีเลชันตามแนวนอน จะแบ่งแต่ละทูเพิลของรีเลชัน r ไปเป็นเป็นหลาย ๆ ส่วน ส่วนการแบ่งรีเลชันตามแนวตั้งจะเป็นการแยกรีเลชัน (decomposition) โดยการแตกโครงสร้างของรีเลชัน r ออกเป็นหลาย ๆ รีเลชันย่อย ในที่นี้จะยกตัวอย่างรีเลชัน Employee ซึ่งมีโครงสร้างของรีเลชันดังนี้

Employee = (empid, name, state, age, salary)

รีเลชัน Employee แสดงได้ดังรูปที่ 13.3

Empid	Name	State	Age	Salary
43125	Nancy	New York	28	4500
43126	Andrew	Texas	26	4250
43129	Janet	New	32	5000

		York		
43135	Margaret	New York	47	8500
43136	Steven	Texas	36	6500
43139	Michael	Texas	24	4000
43140	Robert	Texas	29	4500

รูปที่ 13.3 ตัวอย่างรีเลชัน Employee

1.2.1 Horizontal Fragmentation

รีเลชัน r ถูกแบ่งออกเป็นรีเลชันย่อย ๆ r_1, r_2, \dots, r_n โดยแต่ละทูเปิลของรีเลชัน จะต้องถูกแบ่งออกไปอยู่ใน รีเลชันย่อย ๆ เหล่านั้น

รีเลชันย่อยแต่ละอันจะเป็นส่วนหนึ่งของรีเลชัน r เราใช้สัญลักษณ์ P_i เพื่อแทนรีเลชันย่อย r_i ดังนี้

$$r_i = \sigma_{P_i}(r)$$

ดังนั้น เราสามารถสร้างรีเลชัน r ใหม่โดยใช้วิธีการรวม(Union)ทุก รีเลชันย่อย ได้ดังนี้

$$r = r_1 \cup r_2 \cup \dots \cup r_n$$

จากตัวอย่าง สมมติว่า r ถูกแบ่งออกเป็นรีเลชันย่อย ๆ n รีเลชัน แต่ละรีเลชันย่อย ประกอบด้วยข้อมูลที่เป็นของแต่ละรัฐ จากตัวอย่าง ถ้าในระบบมีรัฐอยู่ 2 รัฐ คือ New York และ Texas จะทำการแบ่งรีเลชันออกเป็น 2 รีเลชัน คือ

$$\text{employee}_1 = \sigma_{\text{state}='New York'}(\text{employee})$$

$$\text{employee}_2 = \sigma_{\text{state}='Texas'}(\text{employee})$$

ทั้ง 2 รีเลชันย่อย แสดงได้ดังรูปที่ 13.4 โดยที่รีเลชันย่อย $employee_1$ จะถูกเก็บไว้ที่เซต New york และ รีเลชันย่อย $employee_2$ จะถูกเก็บไว้ที่เซต Texas

$employee_1$

Empid	Name	State	Age	Salary
43125	Nancy	New York	28	4500
43129	Janet	New York	32	5000
43135	Margaret	New York	47	8500

$employee_2$

Empid	Name	State	Age	Salary
43126	Andrew	Texas	26	4250
43136	Steven	Texas	36	6500
43139	Michael	Texas	24	4000
43140	Robert	Texas	29	4500

รูปที่ 13.4 แสดงการแบ่งรีเลชันตามแนวนอนของรีเลชัน $employee$

1.2.2 Vertical Fragmentation

การแบ่งรีเลชันตามแนวตั้งจะเป็นการแยกแอตทริบิวต์ของรีเลชัน r ออกเป็นรีเลชันย่อยๆ ซึ่งแต่ละรีเลชันย่อยจะประกอบไปด้วยแอตทริบิวต์ของรีเลชัน r นั่นคือ ถ้าเราให้ R เป็นสกีมาของรีเลชัน r และสับเซตของแอตทริบิวต์ R_1, R_2, \dots, R_n ของ R คือ

$$R = R_1 \cup R_2 \cup \dots \cup R_n$$

แต่ละรีเลชันย่อย r_i ของ r คือ

$$r_i = P R_i (r)$$

รีเลชันย่อยที่ได้สามารถนำกลับมาสร้างเป็นรีเลชัน r ได้ใหม่โดยการทำ natural join

$$r = r_1 \bowtie r_2 \bowtie r_3 \bowtie \dots \bowtie r_n$$

ในการทำ natural join เพื่อให้ผลลัพธ์ของการ join ได้เป็นรีเลชัน r เหมือนเดิม สามารถทำได้โดยการใส่แอตทริบิวต์ที่ทำหน้าที่เป็นคีย์หลักของ R เข้าไปในแต่ละ R_i โดยทั่วไปแล้วก็สามารถใช้ซูเปอร์คีย์แทนได้เช่นกัน เพื่อความสะดวกเราจะทำการเพิ่มแอตทริบิวต์พิเศษเข้าไปในรีเลชัน จะเรียกว่า tid โดยที่ค่าของแอตทริบิวต์ tid จะถูกกำหนดให้มีค่าไม่ซ้ำกันเพื่อใช้ในการจำแนกทูเปิลแต่ละทูเปิลออกจากกัน ดังนั้นแอตทริบิวต์ tid จะทำหน้าที่เป็นคีย์คู่แข่งของรีเลชัน r และจะถูกนำไปรวมเข้ากับแต่ละรีเลชันย่อยที่แยกออกมา

พิจารณารีเลชัน employee ทำการเพิ่มแอตทริบิวต์ tid ดังรูปที่ 13.5

Tid	Empid	Name	State	Age	Salary
1	43125	Nancy	New York	28	4500
2	43126	Andrew	Texas	26	4250
3	43129	Janet	New York	32	5000
4	43135	Margaret	New York	47	8500
5	43136	Steven	Texas	36	6500
6	43139	Michael	Texas	24	4000

7	43140	Robert	Texas	29	4500
---	-------	--------	-------	----	------

รูปที่ 13.5 แสดงรีเลชัน employee และแอตทริบิวต์ tid

แสดงการแยกรีเลชันรีเลชัน employee และแอตทริบิวต์ tid ออกเป็น 2 รีเลชันย่อยคือ

employee-schema-1 = (Tid, empid, state)

employee-schema-2 = (Tid, name, age, salary)

รีเลชันย่อยทั้ง 2 ในรูปที่ 13.6 เป็นผลมาจากการดำเนินการดังนี้

employee₁ = P employee-schema-1 (employee)

employee₂ = P employee-schema-2 (employee)

Employee₁

Tid	Empid	State
1	43125	New York
2	43126	Texas
3	43129	New York
4	43135	New York
5	43136	Texas
6	43139	Texas
7	43140	Texas

Employee₂

Tid	Name	Age	Salary
1	Nancy	28	4500
2	Andrew	26	4250
3	Janet	32	5000
4	Margaret	47	8500
5	Steven	36	6500
6	Michael	24	4000
7	Robert	29	4500

รูปที่ 13.6 แสดงการแบ่งรีเลชันตามแนวตั้งของรีเลชัน employee

ดังนั้นเราสามารถสร้างรีเลชัน employee จากรีเลชันย่อยได้จาก

P employee-schema ($employee_1 \bowtie employee_2$)

นิพจน์ $employee_1 \bowtie employee_2$ เป็นรูปแบบพิเศษของการทำ natural join โดยจะใช้แอตทริบิวต์ tid ในการ join ระหว่างรีเลชัน แม้ว่าแอตทริบิวต์ tid จะมีประโยชน์ในการแบ่งรีเลชันตามแนวตั้ง แต่ผู้ใช้จะไม่สามารถมองเห็นแอตทริบิวต์นี้ เนื่องจากเป็นส่วนที่ระบบสร้างขึ้นภายในตัวระบบเองเพื่อจัดการกับรีเลชันย่อยเท่านั้น

1.2.3 Mixed Fragmentation

การแบ่งแบบผสมนี้ รีเลชัน r จะถูกแบ่งออกเป็นรีเลชันย่อย r_1, r_2, \dots, r_n โดยแต่ละรีเลชันเหล่านั้นอาจจะถูกแบ่งนอนหรือแนวตั้งก็ได้จากรีเลชัน r และรีเลชันย่อยแต่ละอันก็สามารถถูกแบ่งตามแนวนอนหรือแนวตั้งต่อไปได้อีก

พิจารณาจากรูปที่ 13.4 รีเลชัน employee ถูกแบ่งตามแนวตั้งเป็นรีเลชันย่อย $employee_1$ และ $employee_2$ จากนั้นเราสามารถที่จะแบ่งรีเลชันย่อย $employee_1$ ตามแนวนอนได้อีก เป็น 2 รีเลชันย่อย ดังนี้

$employee_{11} = s_{state="New\ york"}(employee_1)$

$employee_{12} = s_{state=="Texas"}(employee_1)$

ดังนั้นรีเลชัน r จึงถูกแบ่งออกเป็น 3 รีเลชันย่อย คือ $employee_{11}$, $employee_{12}$ และ $employee_2$ ซึ่งแต่ละ รีเลชันย่อยอาจจะถูกจัดเก็บไว้ต่างไซต์กัน

1.3 Data Replication and Fragmentation

วิธีการนี้เป็นการนำวิธีการทำสำเนาข้อมูล และการแยกข้อมูลมาใช้ร่วมกัน คือรีเลชันย่อยสามารถถูกทำสำเนาได้ และสำเนาของรีเลชันย่อยก็สามารถถูกแยกรีเลชันได้ ตัวอย่างเช่น พิจารณาระบบที่ประกอบไปด้วยไซต์ S_1, S_2, \dots, S_{10} เราสามารถแยกรีเลชัน $employee$ เป็น $employee_{11}$, $employee_{12}$ และ $employee_2$ โดยเราจะเก็บสำเนาของรีเลชันย่อย $employee_{11}$ ไว้ที่ไซต์ S_1, S_2 และ S_3 ทำการสำเนาของรีเลชันย่อย $employee_{12}$ ไว้ที่ไซต์ S_5 และ S_6 และสำเนารีเลชันย่อย $employee_2$ ไว้ที่ไซต์ S_4 , และ S_7 เป็นต้น

ที่มา <http://sot.swu.ac.th/Portals/156/sot/CP342/lesson13/ms1t3.htm>