วิชา ระบบจัดการฐานข้อมูล
(3204 -2004)

หน่วยที่ 7 การเกิดภาวะพร้อมกันและการกู้คืนฐานข้อมูล

Asst. Prof. Juthawut Chantharamalee

Assistant Professor in Computer Science

(Chairperson of B.Sc. Program in Computer Science)

Office. Suan Dusit University, Phone. (+66) 2244-5691

Email. juthawut_cha@dusit.ac.th, jchantharamalee@gmail.com

หน่วยที่ 7 การเกิดภาวะพร้อมกันและการกู้คืนฐานข้อมูล

สาระการเรียนรู้

1. ปัญหาในการใช้ระบบฐานข้อมูล
2. รายการเปลี่ยนแปลง
3. การเกิดภาวะพร้อมกัน
4. ระบบการลงบันทึก
5. การกู้คืนข้อมูล
6. เทคนิคการปิดกั้นข้อมูล

หน่วยที่ 7 การเกิดภาวะพร้อมกันและการกู้คืนฐานข้อมูล

จุดประสงค์การเรียนรู้

1. ขอกถึงปัญหาในการใช้ระบบฐานข้อมูลได้
2. อธิบายวิธีเกิดรายการเปลี่ยนแปลงได้
3. อธิบายการเกิดภาวะพร้อมกันได้
4. อธิบายระบบการลงบันทึกได้
5. อธิบายวิธีการกู้คืนข้อมูลได้
6. ขอกเทคนิคการปิดกั้นข้อมูลได้

1. ปัญหาในการใช้ระบบฐานข้อมูล

ข้อมูลที่จัดเก็บเป็นเอกสารหรือการจัดเก็บด้วยระบบฐานข้อมูลที่ใช้ภาษาคอมพิวเตอร์ในการเขียนชุดคำสั่งเริ่มมีการเขียนด้วยภาษาในยุคที่ 3 เช่น ภาษาฟอร์แทรน ภาษาโคบอต ภาษาซี เป็นกระบวนการเขียนโปรแกรมชุดคำสั่งจะต้องเขียนโดยการใช้โครงสร้างข้อมูลการสร้างแฟ้ม แทรกข้อมูล แก้ไขหรือปรับปรุงข้อมูลอาจเกิดปัญหาหลายๆ อย่าง เช่น

1. ปัญหาในการใช้ระบบฐานข้อมูล

1) ความยุ่งยากจากการประมวลผลกับระบบแฟ้มข้อมูล
2) แฟ้มข้อมูลไม่มีความเป็นอิสระ
3) แฟ้มข้อมูลมีความซ้ำซ้อนมาก
4) แฟ้มข้อมูลมีความถูกต้องของข้อมูลน้อย
5) แฟ้มข้อมูลมีความปลอดภัยน้อย
6) ไม่มีการควบคุมจากศูนย์กลาง

2. รายการเปลี่ยนแปลง

2.1 รายการเปลี่ยนแปลง (Transaction) หมายถึง หน่วยของการทำงานในส่วนของโปรแกรม ที่ถูกเขียนขึ้นมาเพื่อใช้จัดการข้อมูลภายในฐานข้อมูล โดยจะเป็นกลุ่มของการกระทำต่าง ๆ เช่น การเพิ่ม การลบ หรือการปรับปรุงข้อมูล ซึ่งกลุ่มของการกระทำนี้จะต้องถูกทำให้เสร็จสิ้นสมบูรณ์ ถ้ามีเหตุการณ์ได้ทำให้กลุ่มการกระทำนั้นทำงานไม่เสร็จสมบูรณ์ก็จะต้องยกเลิกกลุ่มการกระทำนั้นทั้งหมด

2. รายการเปลี่ยนแปลง

2.2 คำสั่งที่ใช้ในการควบคุมรายการเปลี่ยนแปลง มีดังนี้

2.2.1 COMMIT เป็นคำสั่ง SQL ที่ใช้เพื่อระบุว่ารายการเปลี่ยนแปลงทำงานเสร็จสมบูรณ์ ซึ่ง จะทำการปรับปรุงข้อมูลในหน่วยความจำโดยสมบูรณ์ ดังนั้นในการกระทำรายการเปลี่ยนแปลงจะถือว่า เสร็จสมบูรณ์ได้ก็ต่อเมื่อเจอคำสั่ง COMMIT นั่นเอง

2.2.2 ROLLBACK เป็นคำสั่งแสดงถึงการยกเลิกการกระทำทั้งหมดในรายการเปลี่ยนแปลง ในกรณีที่มีการทำงานบางส่วนของรายการเปลี่ยนแปลงไม่สมบูรณ์ พร้อมทั้งนำข้อมูลเดิมก่อนเริ่มรายการ เปลี่ยนแปลงกลับมา

2.3 คุณสมบัติของรายการเปลี่ยนแปลง (Transaction Properties) รายการเปลี่ยนแปลงควร มีคุณสมบัติดังต่อไปนี้

2. รายการเปลี่ยนแปลง

2.3.1 การเป็นงานเดียว (Atomicity) หมายความว่า ทุกคำสั่งการกระทำที่เกิดขึ้นในรายการ เปลี่ยนแปลงหนึ่งจะต้องทำให้เสร็จสมบูรณ์ ถ้าไม่เสร็จคำสั่ง การกระทำนั้นจะถูกยกเลิกทั้งหมด ดังนั้น รายการเปลี่ยนแปลง 1 รายการจึงเหมือนเป็นการทำงานเพียงงานเดียว เช่น การโอนเงิน 1,000 บาท จาก บัญชี A ไปบัญชี B ถ้ากระบวนการของการโอน คือ

1) อ่านยอดเงินคงเหลือจากบัญชี A เก็บในตัวแปร balance
2) balanceA = balanceA - 1,000
3) อ่านยอดเงินคงเหลือจากบัญชี B เก็บในตัวแปร balanceB
4) balanceB = balanceB + 1,000

การโอนเงิน 1,000 บาทจะทำแค่ขั้นตอนที่ 1 คือ ลดเงินจาก A เลยไม่ได้ ต้องเอาไปเพิ่มใน B ด้วย แต่ถ้าทำขั้นตอนที่ 1, 2 และ 3 เสร็จแล้วแต่พอถึงขั้นตอนที่ 4 คือ เอาเงินไปเพิ่มให้บัญชี B แต่ระบบ หาบัญชี B ไม่เจอ คำสั่งทั้งหมดนี้ก็จะถูกยกเลิกทุกคำสั่ง

2. รายการเปลี่ยนแปลง

2.3.2 การรักษาความถูกต้องของข้อมูล (Consistency) หมายความว่า ทุกรายการเปลี่ยนแปลงต้องรักษาความถูกต้องของข้อมูลในฐานข้อมูลไว้ เช่น จะโอนเงิน 50 บาทจากบัญชี เปลี่ยนแปลงต้องรักษาความถูกต้องของข้อมูลในฐานข้อมูล ไปบัญชี B วิธีการ คือ

1) อ่านยอดเงินคงเหลือจากบัญชี A เก็บในตัวแปร balanceA

2) balanceA = balanceA - 50

3) อ่านยอดเงินคงเหลือจากบัญชี B เก็บในตัวแปร balanceB

4) balanceB = balanceB + 50

2. รายการเปลี่ยนแปลง

ปัญหา คือ ถ้ากำหนดเงื่อนไขว่าห้ามบัญชีใด ๆ มีเงินน้อยกว่า 0 บาท แต่ยอดเงินในบัญชี 4 ที่ถูกเก็บในตัวแปร balanced มีแค่ 10 บาท ถ้าเอา 50 ไปลบมันจะได้ - 40 ซึ่งผิดข้อกำหนด โดยถ้าไม่มีข้อกำหนดนี้ก็ยังมีการตรวจสอบโดยเอายอดเงิน A + ยอดเงิน B ก่อนทำการโอนแล้วเอายอดเงิน A + ยอดเงิน B หลังทำการโอนมาตรวจสอบว่าเท่ากันหรือเปล่าด้วย เพราะถ้าไม่เท่ากันจะทราบว่าไม่ถูกต้อง หรือ การที่พยายามเอาตัวอักษรไปใส่ช่องข้อมูลในฐานข้อมูลที่เก็บตัวเลขก็จะเกิด Error และถ้า Error มันก็จะ Roll Back ย้อนกลับไปตอนก่อนที่จะทำคำสั่ง (Pre-transaction)

2. รายการเปลี่ยนแปลง

2.3.3 การแยกจากรายการเปลี่ยนแปลงอื่น (Isolation) หมายความว่า หากรายการเปลี่ยนแปลงมีความต้องการใช้ข้อมูลร่วมกันในขณะใดขณะหนึ่ง ข้อมูลที่ถูกใช้ระหว่างการประมวลผลรายการ เปลี่ยนแปลงหนึ่งไม่สามารถนำไปใช้กับรายการเปลี่ยนแปลงอื่นได้ จนกว่ารายการเปลี่ยนแปลงที่กำลังใช้ข้อมูลนั้นทำการประมวลผลเสร็จสมบูรณ์ก่อน

2. รายการเปลี่ยนแปลง

2.3.4 การมีสถานะที่มั่นคง (Durability) หมายความว่า การทำรายการเปลี่ยนแปลงเพื่อ ปรับปรุงข้อมูลในฐานข้อมูล ฐานข้อมูลต้องอยู่ในสถานะที่มั่นคง มีความสอดคล้องกันของข้อมูล แม้จะ เกิดความขัดข้องในระหว่างที่มีการทำรายการเปลี่ยนแปลง เช่น จะโอนเงิน 50 บาทจากบัญชี A ไปบัญชี B วิธีการ คือ

1) อ่านยอดเงินคงเหลือจากบัญชี A เก็บในตัวแปร balance

2) balanceA = balanceA - 50

3) อ่านยอดเงินคงเหลือจากบัญชี B เก็บในตัวแปร balanceB

4) balanceB = balanceB + 50

2. รายการเปลี่ยนแปลง

ไม่มีปัญหาในขั้นตอน AGI (Atomicity Consistency Isolation) และเป็น Committed Transaction กำลังจะถูกบันทึกลงใน Harddisk หัวอ่านกำลังวิ่งผ่าน Sector แล้วขณะที่หัวอ่านเขียนข้อมูล กำลังจะเพิ่มข้อมูลลงไปเกิดไฟฟ้าดับ และเมื่อเปิดระบบขึ้นมาได้ก็จะทำ Transaction นี้ต่อให้เสร็จโดยจะ ไม่มีการยกเลิกแล้ว

3. การเกิดภาวะพร้อมกัน

โดยทั่วไปในระบบฐานข้อมูลจะมีผู้ใช้หลายคนใช้ข้อมูลพร้อมกันในบางครั้งรายการต่างๆ ถูกส่งไป ประมวลผลโดยไม่เกิดปัญหา แต่ในบางครั้งก็อาจจะเกิดปัญหาได้เพราะผลของข้อมูลหนึ่งที่กำลังประมวลผล อยู่จำเป็นต้องปรับค่าของข้อมูลนั้น ๆ ให้เสร็จสิ้นก่อนที่จะถูกเรียกไปใช้โดยผู้ใช้คนอื่นๆ ดังนั้นเมื่อเกิดภาวะ ในการเรียกใช้ข้อมูลพร้อมกันจะเป็นผลให้ค่าของข้อมูลที่ผู้ใช้คนอื่นๆ ได้รับไปไม่ถูกต้อง ในหัวข้อนี้จะ กล่าวถึงความหมายของรายการ (Transaction) เทคนิคในการควบคุมการเกิดภาวะพร้อมกัน (Concurrency) และปัญหาที่เกิดจากการควบคุมการเกิดภาวะพร้อมกัน

3. การเกิดภาวะพร้อมกัน

3.1 รายการ

รายการ หมายถึง กลุ่มของคำสั่งงานหรือชุดคำสั่งงานที่ถูกสั่งให้กระทำการ (Execute) กับข้อมูล ของฐานข้อมูลโดยการอ่าน (Read) เช่น การเรียกดูข้อมูลด้วยคำสั่ง SELECT หรือการเขียนข้อมูล (Write) เช่น คำสั่งเพิ่ม (Insert) ลบ (Delete) หรือปรับ (Update) ข้อมูลในฐานข้อมูล เช่น ถ้าหากต้องการเพิ่ม ข้อมูลการส่งสินค้าเดิม (TOTOTY) จากรีเลชัน

3. การเกิดภาวะพร้อมกัน

```
EXEC SQL INSERT
        INTO ORDER (SNO, PNO, QTY)
        VALUES ('S6, 'P11, 100);
EXEC SQL UPDATE
        SET TOTQTY = TOTOTY + 100
        WHERE PNO = 'P1;
        EXEC SQL COMMIT;
```

การเรียกใช้ข้อมูลจากฐานข้อมูลโดยการประมวลผลรายการต่างๆ อาจจะทำสำเร็จหรือไม่ก็ได้ เพื่อให้เข้าใจถึงขั้นตอนของการอ่านหรือการเขียนข้อมูล จะกล่าวถึงขั้นตอนการอ่านและเขียนข้อมูลว่า เมื่อไร ถึงจะเรียกว่ารายการไม่ถูกเปลี่ยนแปลงเสร็จสมบูรณ์ (COMMIT) หรือถูกเปลี่ยนให้เป็นค่าเดิม (ROLLBACK) ดังนี้

3. การเกิดภาวะพร้อมกัน

3.1.1 การอ่านข้อมูล (X) ประกอบด้วยขั้นตอนดังนี้

1) ค้นหาที่อยู่ในบล็อก (Block) ของจานบันทึก (Disk) เพื่อหาข้อมูล X

2) คัดลอกบล็อกของจานบันทึกลงในที่พักข้อมูล (Buffer) ของหน่วยความจำหลัก

3) คัดลอกข้อมูล X จากที่พักข้อมูลไปยังชุดคำสั่งงานที่ใช้ข้อมูล X

3. การเกิดภาวะพร้อมกัน

3.1.2 การเขียนข้อมูล (X) ประกอบด้วยขั้นตอนดังนี้

1) ค้นหาที่อยู่ในบล็อกของจานบันทึกเพื่อหาข้อมูล X
2) คัดลอกบล็อกของจานบันทึกลงในที่พักข้อมูลของหน่วยความจำหลัก
3) คัดลอกข้อมูล X จากชุดคำสั่งงานที่ใช้ข้อมูล X ไปยังที่อยู่ในที่พักข้อมูล
4) นำค่าที่ปรับใหม่ที่อยู่ในที่พักข้อมูลไปยังจานบันทึก

ขั้นตอนที่ 4 ของการเขียนข้อมูลเป็นการปรับเปลี่ยนข้อมูลของฐานข้อมูลที่เก็บไว้ในสื่อเก็บข้อมูล ซึ่งเมื่อเสร็จสิ้นขั้นตอนนี้จึงถือว่ารายการนั้นๆ ได้กระทำการเสร็จสมบูรณ์ (COMMIT) หากมีเหตุขัดข้องหรือข้อผิดพลาดเกิดขึ้นในระบบก่อนที่รายการจะเสร็จสมบูรณ์ ก็จะทำให้รายการนั้นๆ กลับไปสู่จุดเริ่มต้นหรือ เป็นค่าเดิม (ROLLBACK) โดยรายการเปลี่ยนแปลงต่างๆ จะไม่ถูกกระทำ (UNDO)

3. การเกิดภาวะพร้อมกัน

3.2 ปัญหาการเกิดภาวะพร้อมกัน

ในระบบฐานข้อมูลที่มีผู้ใช้หลายคนในระบบและสามารถใช้ข้อมูลในเวลาเดียวกันได้พร้อมกัน ซึ่ง ลักษณะของการทำงานแบบหลายชุดคำสั่งงานในเวลาเดียวกัน อาจก่อให้เกิดปัญหาการเกิดภาวะพร้อมกันได้ หากขาดการควบคุมเพื่อให้เห็นถึงปัญหาที่กล่าวได้ชัดเจนยิ่งขึ้น จะใช้ตัวอย่างระบบข้อมูลการจองตั๋วสายการบิน โดยสมมติให้มีรายการเกี่ยวกับการจองตั๋วเข้ามาพร้อมกันดังนี้

รายการที่ T1 เป็นรายการที่ต้องการยกเลิกการจองที่นั่ง N ซึ่งได้เคยจองที่นั่งอยู่ในรายการของสายการบินชื่อ A ในขณะเดียวกันต้องการจองที่นั่งเดียวกันนี้จากสายการบินอื่นที่มีข้อมูลเก็บไว้ในรายการสายการบินชื่อ B ตัวอย่างของรายการนี้ในการเขียนซูโดโค้ด (Pseudo Code)

3. การเกิดภาวะพร้อมกัน

```
T1:     READ(A)
        UPDATE A = A-N
        WRITE(A)
        READ(B)
        UPDATE B = B+N
        WRITE(B)
```

รายการที่ T1 เป็นรายการที่ต้องการยกเลิกการจองที่นั่ง N ซึ่งได้เคยจองที่นั่งอยู่ในรายการของ สายการบินชื่อ A ในขณะเดียวกันต้องการจองที่นั่งเดียวกันนี้จากสายการบินอื่นที่มีข้อมูลเก็บไว้ในรายการ สายการบินชื่อ B ตัวอย่างของรายการนี้ในการเขียนซูโดโค้ด (Pseudo Code)

3. การเกิดภาวะพร้อมกัน

```
T2:     READ(A)
        UPDATE A = A+M
        WRITE(A)
```

รายการ T2 เป็น เป็นรายการที่ต้องการสำรองที่นั่ง M ของสายการบิน A ตัวอย่างของรายการเป็นดังนี้

3. การเกิดภาวะพร้อมกัน

3.2.1 ปัญหาข้อมูลไม่ได้ถูกปรับให้ถูกต้อง สมมติให้รายการ T1 และ T2 ถูกส่งเข้าในเวลาใกล้เคียงกัน และการประมวลผลของรายการทั้งสองเป็นการประมวลผลแบบสลับกัน (Interleave) ผลลัพธ์สุดท้ายของ A จะไม่ถูกต้อง เพราะ T2 อ่านค่าของ A ก่อนที่ T1 จะเปลี่ยนค่าให้ถูกต้องในฐานข้อมูล

ตัวอย่างเช่น ถ้าค่าเริ่มต้นของการจองที่นั่งของ A = 70 และ B = 81 มีการยกเลิกที่นั่งจากสายการบิน A ไปใช้บริการ B = 5 (N) และมีการสำรองที่นั่งของสายการ A จากรายการ T2) เพิ่มอีก 4 (M) ดังนั้นค่าที่ควรจะเป็นของ A คือ 69 แต่จากตารางที่ 7.1 A มีค่าเท่ากับ 74 ทั้งนี้เพราะ การปรับค่า A จากการยกเลิก 5 ที่นั่ง ไม่ได้ถูกปรับปรุงค่าลงในฐานข้อมูลก่อนที่รายการ T2 จะอ่านค่า A เมื่อเวลา T2

3. การเกิดภาวะพร้อมกัน

ตารางที่ 7.1 ข้อมูลที่ไม่ได้ถูกปรับให้ถูกต้อง

เวลา	รายการ T1	รายการ T2
t1	READ A (70) UPDATE A = A-N (70-5)	
t3		READ A (70) UPDATE A= A+M (70+4)
t3	WRITE A (65) READ B (81)	
t4		WRITE A (74)
t5	UPDATE B = B+N (81+5) WRITE B (86)	

3. การเกิดภาวะพร้อมกัน

3.2.2 ปัญหาข้อมูลถูกปรับเปลี่ยนค่าไม่เสร็จสมบูรณ์ ปัญหานี้เกิดจากขั้นเมื่อการปรับค่าของรายการ ยังทำไม่เสร็จสมบูรณ์และเกิดปัญหาบางอย่างขึ้น เช่น ระบบล้มเหลวก่อนการปรับค่าใหม่ ให้ถูกต้อง ทำให้ข้อมูลนั้นยังเป็นค่าเดิม ซึ่งในกรณีที่มีกรายการหนึ่งต้องใช้ค่าของข้อมูลนั้น จะได้รับค่าที่ไม่ถูกต้อง ดังตารางที่ 7.2

3. การเกิดภาวะพร้อมกัน

ตารางที่ 7.2 ข้อมูลที่ถูกปรับเปลี่ยนค่าไม่เสร็จสมบูรณ์

เวลา	รายการ T1	รายการ T2
t1	READ A (70) UPDATE A = A-N (70-5) ROLLBACK รายการ T1 ทำไม่เสร็จ เนื่องจากระบบขัดข้อง ทำให้ค่า A เปลี่ยนค่าไปเป็นค่าเดิมและรายการ T2 นำค่าเดิม (70) ไปใช้	
t2		READ A (70) UPDATE A= A+M (70+4)

3. การเกิดภาวะพร้อมกัน

3.2.3 ปัญหาการวิเคราะห์ผลลัพธ์ของข้อมูลไม่ถูกต้อง ปัญหาสองข้อแรกที่กล่าวมาเป็นปัญหาที่เกิดจากการปรับปรุงข้อมูล (Update) ส่วนปัญหาในข้อนี้เป็นปัญหาที่เกิดขึ้นกับรายการที่อ่านข้อมูล แต่ค่าที่อ่านอยู่ในระหว่างการปรับปรุงข้อมูลอยู่ สมมติให้รายการ T2 เป็นการคำนวณหาค่าการจองที่นั่งรวมทั้งหมด (SUS A-E) ในขณะเดียวกันรายการ T1 ก็ถูกส่งเข้าทำงานในเครื่องเช่นเดียวกัน ซึ่งเป็นการปรับปรุงค่า A คือ 65 และค่า B คือ 86 ผลลัพธ์รวมที่ถูกต้อง คือ 151 แต่รายการ T2 ได้อ่านค่าของ A ซึ่งเป็นค่าก่อนที่ จำนวนที่นั่งจะถูกยกเลิก (N) ทำให้ค่าที่นำไปคำนวณผลลัพธ์รวมของการจองที่นั่งไม่ถูกต้อง (156) ซึ่งกรณีนี้การรวมค่า A และ B ควรจะทำเมื่อ T1 ทำเสร็จก่อน

3. การเกิดภาวะพร้อมกัน

ตารางที่ 7.3 ปัญหาการอ่านค่าข้อมูลระหว่างการปรับปรุง

เวลา	รายการ T1	รายการ T2
t1		READ A
t2	READ A UPDATE A = A-N (70-5) WRITE A (65)	
t3	READ B UPDATE B = B+N (81+5) WRITE B (86)	
t4		READ B SUM A+B (70+86) ซึ่งไม่ถูกต้อง

3. การเกิดภาวะพร้อมกัน

3.3 ลำดับขั้น

ปัญหาดังกล่าวข้างต้นสามารถจัดการได้ ด้วยการสร้างกฎเกณฑ์ที่ใช้ในการควบคุมการเกิดภาวะ พร้อมกัน (Concurrency Control Protocol) เพื่อกำหนดลำดับให้แต่ละรายการสามารถกระทำสลับกัน โดยไม่เกิดปัญหาความไม่ถูกต้องของข้อมูล

กฎเกณฑ์ต่าง ๆ เพื่อแก้ไขความไม่ถูกต้องของข้อมูลที่เกิดจากการเกิดภาวะพร้อมกัน คือ การลำดับ (Serializability) ด้วยการจัดลำดับในการอ่าน (Read) หรือเขียน (Write) ของรายการสองรายการที่เกิดขึ้นพร้อมกันโดยยึดหลักดังนี้

3. การเกิดภาวะพร้อมกัน

3.3.1 ถ้ารายการสองรายการเป็นการอ่านข้อมูลจะไม่กระทบกับฐานข้อมูล ดังนั้นจึงไม่มี ความจำเป็นที่ต้องจัดลำดับของรายการทั้งสอง

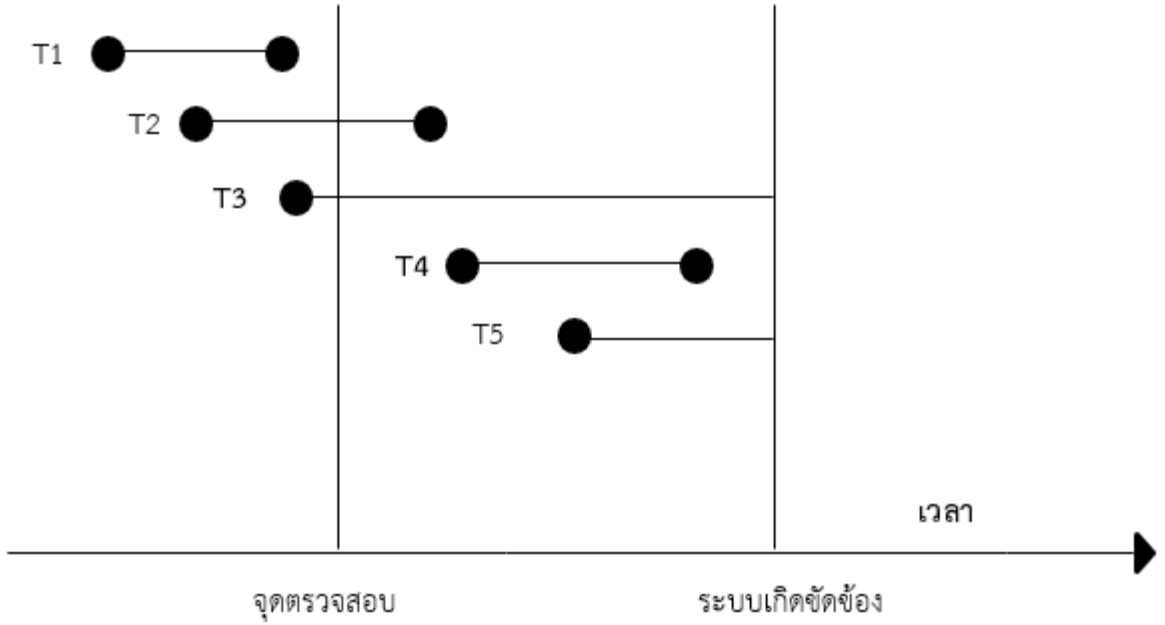
3.3.2 ถ้ารายการสองรายการเป็นการอ่านหรือเขียนรายละเอียดของข้อมูลที่ไม่เหมือนกัน และ ไม่ก่อให้เกิดข้อมูลที่ขัดแย้งกัน ดังนั้นจึงไม่มีความจำเป็นที่ต้องจัดลำดับของรายการทั้งหมด

3.3.3 ถ้ารายการหนึ่งเป็นการเขียนรายละเอียดของข้อมูลโดยที่อีกรายการหนึ่งอ่านหรือต้องเขียนรายละเอียดของข้อมูลเดียวกัน ซึ่งอาจก่อให้เกิดข้อมูลที่ขัดแย้งกัน ดังนั้นจึงจำเป็นต้องลำดับการกระทำของรายการทั้งสองตามลำดับก่อนหลัง

4. ระบบการลงบันทึก

ระบบการลงบันทึกรายการที่ทำการบันทึกค่าของข้อมูลทั้งก่อนและหลังการแก้ไขลงใน Log File หรือ Journal การลงบันทึกจะเริ่มเมื่อมีการกระทำเกิดขึ้นกับรายการ และจะบันทึกรายการที่มีผลต่อการเปลี่ยนแปลงข้อมูลในฐานข้อมูลจึงอ่านค่าใหม่เพื่อบันทึกลงในฐานข้อมูล ณ จุดนั้นถือได้ว่ารายการนั้นได้ถูกกระทำจนเสร็จสมบูรณ์แล้ว (Commit) การใช้เทคนิคของจุดตรวจสอบ (Check Point) ซึ่งระบบจะกำหนดช่วงเวลาที่จะทำจุดตรวจสอบขึ้นเพื่อบันทึกข้อมูลลงในฐานข้อมูล และบันทึกเวลาของจุดตรวจสอบลงใน Log File หากรายการใดทำเสร็จก่อน ถึงจุดตรวจสอบแล้วเกิดความผิดพลาดของระบบรายการนั้นจะไม่ทำใหม่ (Redo) รายการใดที่ยังทำไม่เสร็จแต่ระบบเกิดขัดข้อง รายการนั้นจะถูก (Undo) ส่วนรายการที่ต้องทำใหม่ก็คือรายการที่ยังทำไม่เสร็จ ขณะที่มีการบันทึกรายการที่ผ่านจุดตรวจ แต่รายการนั้นได้ถูกทำเสร็จก่อนที่ระบบจะเกิดความขัดข้องขึ้นดังภาพที่ 7.1

4. ระบบการลงบันทึก



ภาพที่ 7.1 รายการเปลี่ยนแปลงในช่วงเวลาที่ทำการตรวจสอบ

4. ระบบการลงบันทึก

จากภาพที่ 7.1 จะเห็นได้ว่า

1) รายการ T1 ได้ถูกกระทำเสร็จสมบูรณ์ก่อนถึงจุดตรวจสอบ

2) รายการ T2 เกิดขึ้นก่อนจุดตรวจสอบและเสร็จสิ้นก่อนระบบเกิดความขัดข้อง

3) รายการ T3 เกิดขึ้นก่อนจุดตรวจสอบและยังทำไม่เสร็จ เมื่อระบบเกิดความขัดข้อง

4) รายการ T4 เกิดขึ้นหลังจุดตรวจสอบและเสร็จสิ้นก่อนระบบเกิดความขัดข้อง

5) รายการ T5 เกิดขึ้นหลังจุดตรวจสอบและยังทำไม่เสร็จ เมื่อระบบเกิดความขัดของ

หากพบรายการลักษณะนี้เกิดขึ้นจึงควรพิจารณาดังนี้

1) รายการ T1 จะไม่ต้องทำใหม่ เพราะได้บันทึกรายการเปลี่ยนแปลงลงในฐานข้อมูลเรียบร้อยแล้ว

2) รายการ T2 และ T4 จะต้องทำซ้ำใหม่ (Redo)

3) รายการ T3 และ T5 ต้องถูกยกเลิก (Undo) และกลับเป็นค่าเริ่มต้นเพราะรายการทั้งสองยังทำไม่เสร็จ

5. การกู้คืนข้อมูล

การเกิดภาวะขัดข้องนั้นอาจเนื่องมาจากสาเหตุที่ระบบเกิดความขัดข้อง หรือเกิดจากสาเหตุความผิดพลาดของสื่อบันทึกข้อมูล ซึ่งการกู้ข้อมูลที่เกิดจากสาเหตุภาวะขัดข้องที่ต่างกัน ก็จะใช้วิธีการแก้ปัญหาที่ต่างกันด้วย ในหัวข้อนี้จะกล่าวถึงชนิดของการกู้คืน ได้แก่

5.1 การกู้คืนที่เกิดจากระบบเกิดความผิดพลาด (System Recovery) เป็นความขัดข้องที่มีผลต่อข้อมูลที่อยู่ในหน่วยความจำขณะที่มีการประมวลผล วิธีจะนำมาใช้ในการกู้คืน คือ การนำระบบการลงบันทึกดังกล่าวข้างต้นมาใช้

5. การกู้คืนข้อมูล

5.2 การกู้คืนที่เกิดจากความผิดพลาดของสื่อบันทึก (Media Recovery) เป็นความขัดข้องที่มีผลทำให้ข้อมูลบนสื่อบันทึกเสียหายหรือถูกทำลาย วิธีที่จะนำมาใช้ในการก็คือการทำการสำรองข้อมูล (Backup) ไว้ตลอดเวลา เมื่อเกิดความขัดข้องกับสื่อบันทึกข้อมูลที่ใช้งานอยู่ ก็จะนำข้อมูลสำรองที่ทำไว้ล่าสุดถ่ายลงในสื่อบันทึกชุดใหม่ และใช้ระบบการลงบันทึกมาช่วยเพื่อพิจารณารายการที่ต้องทำใหม่ซึ่งเกิดหลังจากทำการสำรองข้อมูลครั้งสุดท้าย

6. เทคนิคการปิดกันข้อมูล

เทคนิคการปิดกั้น (Lock Technique) ซึ่งจะกล่าวรายละเอียดดังต่อไปนี้

6.1 ชนิดของการปิดกั้น มี 2 ชนิด คือ

6.1.1 การปิดกั้นแบบร่วม (Shared Lock หรือ Read Lock หรือ S Lock) เป็นการปิดกั้น ที่ยอมให้ผู้ใช้หลายคนสามารถทำการอ่านรายการหนึ่งๆ ร่วมกันได้แต่ทำการแก้ไขไม่ได้

6.1.2 การปิดกันแบบเฉพาะ (Exclusive Lock หรือ Write Lock หรือ X Lock) เป็น ปิดกั้นที่ทำได้โดยผู้ใช้หนึ่งคนเท่านั้น เมื่อผู้ใช้นั้นต้องการทำการปิดกั้นแบบเฉพาะเพื่อการเปลี่ยนแปลงข้อง ผู้ใช้อื่นจะทำการเปลี่ยนแปลงหรือแก้ไขรายการนั้นไม่ได้จึงต้องคอยจนกว่าจะมีการปล่อยการปิดกั้น

6. เทคนิคการปิดกันข้อมูล

6.2 ระดับของการปิดกัน (Levels of Locking หรือ Granularity) หมายถึง จำนวนข้อมลง จะทำการปิดกั้นได้ในแต่ละครั้ง ซึ่งระบบจัดการฐานข้อมูลแต่ละชนิดอาจจะมีระดับการปิดกันที่แตกต่างกันดังนี้

6.2.1 การปิดกั้นทั้งฐานข้อมูล (Database Lock) เป็นการปิดกั้นการใช้ข้อมูลทั้งฐานข้อมูล ซึ่งการปิดกั้นลักษณะนี้อาจจะเกิดขึ้นน้อยมากยกเว้นในระหว่างที่มีการทำสำรอง (Backup) ฐานข้อมูลทั้งหมด

6. เทคนิคการปิดกันข้อมูล

6.2.2 การปิดกั้นทั้งตาราง (Table Lock) เป็นการปิดกั้นการใช้ข้อมูลทั้งตารางซึ่งการปิดกั้น นี้จะทำเมื่อมีการปรับปรุงข้อมูลทั้งตาราง เช่น การปรับโครงสร้างบริษัทใหม่ทั้งรหัสและชื่อแผนก เป็นต้น

6.2.3 การปิดกั้นเนื้อที่บางส่วนของหน่วยความจำ (Block Page หรือ Database Space) เป็นการปิดกั้นพื้นที่ที่เก็บฐานข้อมูลจริงของข้อมูล (Physical Storage Block) ของรายการหนึ่ง

6. เทคนิคการปิดกันข้อมูล

6.2.4 การปิดกั้นบางแถว (Row หรือ Record Lock) เป็นการปิดกั้นการใช้ข้อมูลเพียงบาง ทูเพิลหรือบางแถวจากตารางหนึ่ง ๆ เท่านั้น ข้อมูลทูเพิลอื่นที่ไม่เกี่ยวข้องก็จะสามารถใช้งานโดยผู้ใช้คนอื่นได้

6.2.5 การปิดกั้นบางคอลัมน์ (Column Lock) เป็นการปิดกั้นเฉพาะบางคอลัมน์ของข้อมูล กรณีนี้จะใช้กับคอลัมน์ที่มีการปรับปรุงข้อมูลบ่อยๆ ในขณะที่คอลัมน์อื่นไม่ค่อยมีการปรับปรุงข้อมูล เช่น ข้อมูลสินค้าคงคลัง คอลัมน์ที่แสดงยอดคงเหลือจะมีการปรับปรุงค่าบ่อย ในขณะที่คอลัมน์อื่นไม่ค่อยจะมี การปรับปรุงข้อมูลมาก เช่น คำอธิบายลักษณะสินค้า สถานที่จัดเก็บ เป็นต้น

6. เทคนิคการปิดกันข้อมูล

การปิดกั้นข้อมูลในรูปแบบต่างๆ สามารถใช้คำสั่งที่ใช้ในระบบจัดการฐานข้อมูลสั่งให้ทำการปิดกั้น ตามประเภทที่ต้องการ ทั้งนี้การที่ระบุว่าจะมีการปิดกั้นแบบใดนั้น ผู้จัดการฐานข้อมูลต้องศึกษาถึงการเรียกใช้ข้อมูลว่ามีโอกาสจะเกิดปัญหาการเกิดภาวะพร้อมกันอย่างไรบ้าง คำสั่งใน SQL ที่ใช้ในการปิดกั้น คือ

LOCK TABLE

```
รูปแบบ LOCK TABLE <TABLE NAME > IN <SHARE |EXCLUSIVE > MODE;
```

เช่น การปิดกั้นรีเลชัน EMPLOYEE แบบร่วมคำสั่งที่ใช้ คือ

```
LOCK TABLE EMPLOYEE IN SHARE MODE;
```

6. เทคนิคการปิดกันข้อมูล

6.3 การปิดกั้นแบบสองระยะ (2PL: Two Phase Locking) ในบางครั้งเทคนิคการปิดกันทั่วไป รายการเกิดภาวะพร้อมกันและไม่แน่ใจว่าเกิดการลำดับที่เหมาะสม (Serializability) จะเพื่อให้แน่ใจว่าไม่เกิดภาวะพร้อมกันที่จะส่งผลให้ข้อมูลไม่ถูกต้อง คือ การใช้เทคนิค
เองระยะ ซึ่งมีกฎเกณฑ์ว่าการกระทำของรายการหนึ่งจะประกอบด้วย 2 ระยะ คือ

6.3.1 Growing Phase เป็นช่วงเวลาที่รายการหนึ่งจะต้องระบุถึงการปิดกั้นที่จะใช้ ซึ่งอาจจะ ลดกันแบบร่วมหรือแบบเฉพาะได้และเป็นระยะที่ยังไม่ปลดการปิดกั้น

6.3.2 Shrinking Phase เป็นช่วงเวลาที่รายการหนึ่งจะต้องปลดปล่อยการปิดกั้นและจะมีการปิดกันอีกไม่ได้

ตารางที่ 7.4 การปิดกั้นการปรับปรุงข้อมูล

เวลา	T1	T2	X
T1	Begin_Transaction		100
T2	Write Lock (X)	Begin_Transaction	100
T3	Read (X)	Write Lock (X)	100
T4	Balance = X+100	Wait	100
T5	Write X	Wait	200
T6	Unlock (X)	Wait	200
T7	Commit	Read X	200
T8		Balance = X-10	200
T9		Write X	190
T10		Unlock X	190
T11		Commit	190

จากตารางที่ 7,4 T1 ได้ทำการปิดกันแบบเฉพาะเพื่อปรับปรุงข้อมูลยอดคงเหลือของ X (Growing Phase) ด้วยการอ่านค่า X และเพิ่มยอดเงินฝากอีก 100 ในขณะเดียวกัน T2 การขอปิดกั้นแบบเฉพาะเมื่อเวลา T3 เพื่อปรับปรุงยอดคงเหลือของ X จากการถอนอีก 10 เมื่อเป็น การขอปิดกั้นจะไม่ได้รับอนุมัติทันทีแต่ต้องรอจนกว่า T1 ปลดปล่อยการปิดกันและเมื่อ T1 ทำเสร็จก็จะ ปลดการปิดกั้น (Shrinking Phase) หลังจากนั้น T2 ก็จะสามารถทำการปิดกันและทำรายการของ T2 ได้

6. เทคนิคการปิดกันข้อมูล

6.4 ปัญหาที่เกิดจากการปิดกั้น ถึงแม้ว่าเทคนิคการปิดกันจะช่วยแก้ปัญหาการเกิดภาวะพร้อม ไม่ว่าจะเป็นปัญหาข้อมูลไม่ได้ถูกปรับให้ถูกต้อง ปัญหาข้อมูลถูกปรับเปลี่ยนค่าไม่เสร็จสมบูรณ์ หรือปัญหา การวิเคราะห์ผลลัพธ์ของข้อมูลไม่ถูกต้องตามที่ได้กล่าวมาในตอนต้น ปัญหาที่อาจจะเกิดจากการนำเทคนิค การปิดกั้นมาใช้ คือ การปิดตายของรายการ (Deadlock) การปิดตายนี้จะเกิดขึ้นเมื่อมีรายการอย่างน้อย สองรายการต่างหยุดรอซึ่งกันและกัน เช่น จากตารางที่ 7.5 รายการ T1 และ T2 ต่างหยุดรอซึ่งกันและกัน

6. เทคนิคการปิดกันข้อมูล

ตารางที่ 7.5 ปัญหาที่เกิดจากการปิดกั้น

เวลา	รายการ T1	รายการ T2
t1	1. ปิดกั้น Record X	
t2		1. ปิดกั้น Record Y
t3	2. เรียกใช้ Record Y	
t4		2. เรียกใช้ Record X (รอ Record x)
t5	(รอ Record Y)	

6. เทคนิคการปิดกันข้อมูล

วิธีแก้ปัญหาการปิดตาย การแก้ปัญหาการปิดตาย (Deadlock) อาจทำได้ 2 ลักษณะ คือ

1) แบบป้องกันไว้ล่วงหน้า (Deadlock Prevention) เป็นการป้องกันโดยที่มีการเขียนคำสั่งเพื่อทำการปิดกั้นไว้ล่วงหน้า จากตารางที่ 7.5 รายการ T1 และรายการ T2 จะต้องทำการปิดกั้นก่อนการประมวลผล หากมีข้อมูลใดที่ถูกปิดกั้น โปรแกรมจะต้องรอจนกว่าจะมีการปล่อยการปิดกั้น

การป้องกันในลักษณะนี้ค่อนข้างจะทำได้ยาก เพราะเป็นการยากที่จะทราบล่วงหน้าว่าจะมีผู้ใช้หลายคนมาเรียกข้อมูลพร้อมกันในเวลาใด

6. เทคนิคการปิดกันข้อมูล

2) แบบแก้ไขปัญหาเมื่อเกิดการปิดตาย (deadlock resolution) วิธีการนี้ยอมให้มีการปิดตาย เกิดขึ้น แต่ระบบจัดการฐานข้อมูลจะสร้างกลไกเพื่อแก้ไขปัญหาการปิดตายที่เกิดขึ้น เช่น ระบบจะจัดทำตารางแสดงให้ทราบว่าใครเป็นผู้เรียกใช้ข้อมูลอะไร เมื่อไหร่ เพื่อจะได้ปลดการปิดตายจากรายการใดรายการหนึ่ง เทคนิคที่นิยมในการแก้ไขการแก้ปัญหานี้คือ WFG (Wait For Graph) เพื่อดูว่ารายการไหนสัมพันธ์กันบ้างการปิดตายเกิดขึ้นเมื่อรูปของ WFG เป็นวงกลม เช่น รายการ T1 รอค่า x จาก T2 รอค่า X จาก T2 และรายาการ T2 รอค่าจาก Y จาก T1