

การกู้ข้อมูลจากความขัดข้องที่เกิดจากข้อผิดพลาดของรายการและสื่อบันทึกข้อมูล

ปัญหาของความขัดข้องของระบบจะทำให้เกิดสถานะของทรานแซกชันได้ 2 สถานะคือ ทรานแซกชันยังทำไม่เสร็จสมบูรณ์ หรือ ทรานแซกชันทำเสร็จแล้วแต่ยังไม่ได้บันทึกลงฐานข้อมูล การกู้ข้อมูลหรือการฟื้นฟูสภาพไม่ว่าจะเกิดความขัดข้องประการใดๆจะใช้วิธีการฟื้นฟูสภาพแบบล็อกเบส(logbase recovery) โดยไฟล์ประวัติ(log file) จะเข้ามาช่วยในการนำข้อมูลกลับคืนมาให้อยู่ในภาวะที่ถูกต้อง

1. ลักษณะความขัดข้องที่เกิดจากข้อผิดพลาดของรายการ

ความขัดข้องที่เกิดจากข้อผิดพลาดของรายการ ความขัดข้องของทรานแซกชันมี 2 ลักษณะ คือ

1.1 ความขัดข้องที่สามารถป้องกันได้โดยการตรวจสอบภายในโปรแกรม เช่น ทรานแซกชันการโอนเงิน ถ้ายอดเงินโอนมีค่ามากกว่ายอดเงินที่มีอยู่ในบัญชี เมื่อโปรแกรมตรวจพบจะแสดงข้อความผิดพลาด อาทิ “ยอดเงินไม่เพียงพอ (insufficient fund)” ซึ่งลักษณะนี้ต้องทำการยกเลิกการเปลี่ยนแปลงต่าง ๆ โดยปรับข้อมูลให้เป็นค่าเดิมเมื่อเริ่มต้น หรือ ผู้เขียนโปรแกรมกำหนดขอบเขตของทรานแซกชันไม่ถูกต้อง ดังตัวอย่างที่ความขัดข้องดังกล่าวสามารถป้องกันได้โดยผู้เขียนโปรแกรมต้องตรวจสอบความถูกต้องภายในโปรแกรมให้ครอบคลุมทุกกรณีที่มีโอกาสจะละเมิดกฎควบคุมความถูกต้อง และต้องกำหนดขอบเขตของทรานแซกชันให้ถูกต้อง ตัวอย่างที่ 2 ทรานแซกชันการโอนเงินจากบัญชี A ไปยังบัญชี B โดยกฎควบคุมความถูกต้อง (integrity rule) ในการโอนเงิน คือ ยอดเงินต้องไม่หายไปจากระบบ โดยกำหนดขอบเขตของทรานแซกชันดังนี้.

Begin Tx

```
UPDATE account SET AMOUNT = AMOUNT - x
```

```
WHERE ACCT# = 'ACCT_A';
```

End Tx

Begin Tx

```
UPDATE account SET AMOUNT = AMOUNT + x
```

```
WHERE ACCT# = 'ACCT_B';
```

End Tx

จะเห็นว่าทรานแซกชันถอนเงิน กับ ทรานแซกชันฝากเงิน เป็นคนละทรานแซกชันกัน ซึ่งเป็นการกำหนดขอบเขตของทรานแซกชันที่**ผิด** ซึ่งที่ถูกต้องแล้วควรกำหนดทรานแซกชันถอน และ ทรานแซกชันฝาก อยู่รวมกันเป็นทรานแซกชันเดียว เพราะถ้าอยู่แยกกัน ขณะที่ทรานแซกชันถอนเงินจากบัญชี A ทำเสร็จเรียบร้อยแล้ว แต่ถ้าทรานแซกชันฝากเงินเข้าบัญชี B ผิดพลาด จะทำให้ไปยกเลิกส่วนทรานแซกชันถอนเงินไม่ได้ เนื่องจากการทำงานของทรานแซกชันชุดนี้ทำแยกกัน ดังนั้นจึงมีส่วนของทรานแซกชันถอนเงินสำเร็จเกิดขึ้น และมีส่วนของทรานแซกชันฝากเงินผิดพลาด (failed) เกิดขึ้น ทรานแซกชันทั้งชุดนี้ไม่ได้ถูกกระทำเป็นหน่วยเดียว ลักษณะนี้ถือว่าทรานแซกชันนี้ไม่มีคุณสมบัติอะตอมมิกซิตี้ (atomicity) เพราะที่ผู้เขียนโปรแกรมกำหนดขอบเขตของทรานแซกชันไม่ถูกต้อง นอกจากนี้ผลที่ตามมาทำให้ข้อมูลในฐานข้อมูลไม่มีความถูกต้องเกิดขึ้น ซึ่งทำให้ทรานแซกชันขาดคุณสมบัติของความถูกต้อง (correctness) ซึ่งคุณสมบัติความถูกต้อง (correctness) เป็นหน้าที่ของโปรแกรมเมอร์ที่ต้องรับผิดชอบกำหนดขอบเขตของ ทรานแซกชันให้ถูกต้องโดยทำให้เป็นอะตอมมิกดังนี้

Begin Tx

```
UPDATE account SET AMOUNT = AMOUNT - x
```

```
WHERE ACCT# = 'ACCT_A';
```

```
UPDATE account SET AMOUNT = AMOUNT + x
```

```
WHERE ACCT# = 'ACCT_B';
```

End Tx

1.2 ความขัดข้องที่ไม่สามารถป้องกันได้โดยการตรวจสอบภายในโปรแกรม เช่น การล้น (overflow) เกินพื้นที่ในการทำงาน ซึ่งทำให้มีเนื้อที่ไม่เพียงพอต่อการปฏิบัติการของทรานแซกชัน ซึ่งอาจก่อให้เกิดความผิดพลาดแก่ทรานแซกชันได้

2.การกู้ข้อมูลจากความขัดข้องที่เกิดจากรายการ

การกู้ข้อมูลหรือการฟื้นฟูสภาพจากความขัดข้องของทรานแซกชัน(transaction failures) จะพิจารณาในกรณีที่โปรแกรมไม่สามารถจัดการหรือควบคุมได้ เช่น การล้น (overflow) เกินพื้นที่ในการทำงาน ซึ่งทำให้มีเนื้อที่ไม่เพียงพอต่อการทำงานของทรานแซกชัน วิธีการฟื้นฟูสภาพในกรณีนี้ คือ ยกเลิก (UNDO) การกระทำต่าง ๆ ที่เกิดขึ้นกับทรานแซกชันทั้งหมด และปรับค่าข้อมูลต่าง ๆ ในฐานข้อมูลให้กลับคืนมาเหมือนว่าทรานแซกชันนั้น

ยังไม่เริ่มต้นทำงาน ระบบจัดการฐานข้อมูลจะมีส่วนจัดการการฟื้นฟูสภาพ (Recovery Manager; RM) ทำหน้าที่ดูแลและจัดการฟื้นฟูสภาพ โดยจะตรวจสอบข้อมูลและดำเนินการย้อนหลัง (backward trace) ตามรายละเอียดที่บันทึกในไฟล์ประวัติ (log file) ซึ่งบันทึกค่าตัวแปรต่าง ๆ ทั้งก่อน และหลังการเปลี่ยนแปลงไว้ ทำให้ส่วนจัดการการฟื้นฟูสภาพ สามารถปรับค่าข้อมูลต่าง ๆ กลับคืนมาเป็นค่าเดิมเมื่อเริ่มต้นทรานแซกชันได้

3. การกู้ข้อมูลจากความขัดข้องที่เกิดจากสื่อบันทึกข้อมูล

การฟื้นฟูสภาพจากความขัดข้องของสื่อข้อมูล (Media Failures) ความขัดข้องที่เกิดกับสื่อข้อมูล เช่นบางส่วนของหน่วยความจำสำรองถูกทำลาย ซึ่งอาจเกิดจากหัวอ่านกระทบกับแผ่นดิสก์ ซึ่งอาจมีผลไปทำลายข้อมูลในสื่อข้อมูลได้ การฟื้นฟูสภาพในกรณีนี้ คือ

- จำเป็นต้องอาศัยข้อมูลสำรองครั้งล่าสุด ซึ่งควรจะมีการสำรองข้อมูลไว้อย่างสม่ำเสมอ
- ใช้ข้อมูลในไฟล์ประวัติทำการทำซ้ำ (REDO) สำหรับทรานแซกชันที่ได้กระทำเสร็จสมบูรณ์แล้ว คือทรานแซกชันนั้นคอมมิต (COMMIT) แล้ว

สำหรับความขัดข้องในกรณีนี้ ทำให้เห็นว่าการสำรองข้อมูลมีความจำเป็นอย่างมาก หากมีการสำรองข้อมูลบ่อยๆ ก็ยิ่งทำให้การฟื้นฟูสภาพรวดเร็วมากขึ้น เพราะข้อมูลสำรองมีความทันสมัยใกล้เคียงข้อมูลที่ถูกทำลาย อย่างไรก็ตามการสำรองข้อมูล มักต้องใช้เวลาานาน บางครั้งอาจใช้เวลาเป็นวัน ๆ หากเป็นระบบใหญ่ ๆ ทำให้ไม่สามารถใช้ระบบในงานด้านอื่นๆ ได้ จึงควรกำหนดช่วงเวลาการสำรองข้อมูลให้เหมาะสม เช่นหากมีการเปลี่ยนแปลงโครงสร้างข้อมูลระบบใหม่ก็ควรจะมีการสำรองข้อมูลทันที เป็นต้น

ที่มา <http://sot.swu.ac.th/Portals/156/sot/CP342/lesson11/ms1t3.htm>