แนวคิดเกี่ยวกับการใช้ข้อมูลพร้อมกัน

คุณสมบัติที่สำคัญอย่างหนึ่งของฐานข้อมูลก็คือ การที่ผู้ใช้จากส่วนต่างๆ สามารถจะเรียกใช้ข้อมูลในฐานข้อมูลได้พร้อมๆ กัน ในกรณีที่ฐานข้อมูลที่จัดทำขึ้นในเครื่องไมโครคอมพิวเตอร์เพื่อใช้งานส่วนบุคคลนั้นก็ย่อมไม่จำเป็นต้องคำนึงถึงการควบคุมการทำงานที่เกิดจากผู้ใช้หลายๆ คนที่ต้องการใช้งานฐานข้อมูลนั้นในขณะเดียวกัน เนื่องจากการใช้งานฐานข้อมูลอาจจะเป็นการใช้งานโดยผู้ใช้คนใดคนหนึ่ง ณ เวลาใดเวลาหนึ่งเท่านั้น แต่สำหรับฐานข้อมูลที่ใช้งานในระดับองค์กรที่เป็นฐานข้อมูลในระบบที่ซับซ้อนมากขึ้นนั้น ระบบจัดการฐานข้อมูลจะต้องมีกลไกในการจัดการเพื่อให้ผู้ใช้จากหน่วยงานต่างๆ สามารถเรียกใช้ข้อมูลในฐานข้อมูลเพื่อทำงานได้พร้อมๆ กัน โดยที่ผลลัพธ์ที่ได้จากการทำงานจะต้องถูกต้องเสมอ

1. ความหมายของภาวะพร้อมกัน

คำว่า “ภาวะพร้อมกัน (concurrency)” หมายความว่า การที่มีทรานแซกชันหลายๆ ทรานแซกชันต้องการเรียกใช้ข้อมูลเดียวกันในเวลาเดียวกันจากฐานข้อมูลเพื่อทำงานของแต่ละทรานแซกชัน ภาวะการทำงานพร้อมกันเกิดจากระบบการทำงานได้ 2 ระบบ

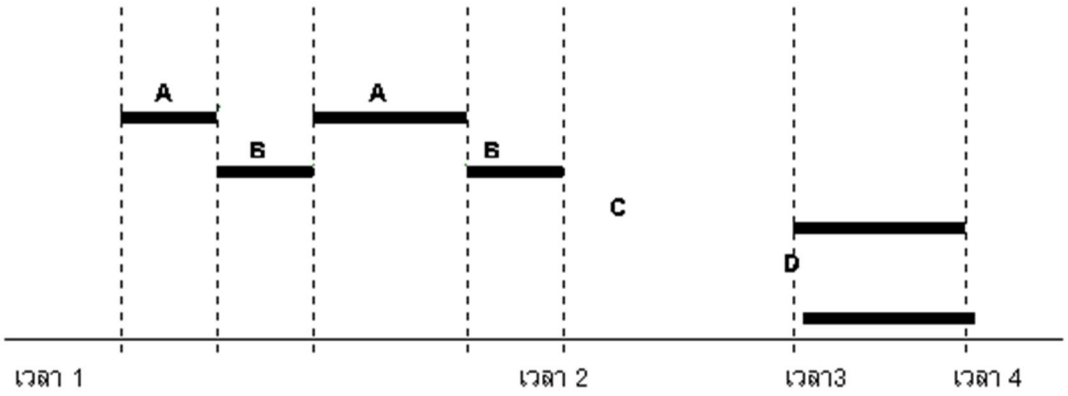
การทำงานในระบบหลายโปรแกรม

การทำงานในระบบการประมวลผลในเวลาเดียวกัน

1.1 การทำงานในระบบหลายโปรแกรม หรือการทำงานแบบมัลติโปรแกรมมิง (multiprogramming) เป็นการทำงานของระบบคอมพิวเตอร์ที่ออกแบบเพื่อให้หน่วยประมวลผลผลกลางหรือซีพียู (Central Processing Unit; CPU) ทำงานหลายๆ งานในขณะเดียวกันได้ ทั้งนี้ด้วยเหตุผลเพื่อให้การใช้งานซีพียูเป็นไปอย่างคุ้มค่าโดยไม่ต้องอยู่ว่าง (idle) เนื่องจากซีพียูจะมีความเร็วในการทำงานสูงกว่าอุปกรณ์อื่นๆ ถ้าหากไม่มีระบบการทำงานแบบมัลติโปรแกรมมิง ซีพียูต้องทำงานใดงานหนึ่งจนเสร็จจึงจะสามารถทำงานที่ 2, 3ต่อไปได้ ซึ่งหมายความขณะที่งานนั้นกำลังใช้เครื่องพิมพ์หรืออุปกรณ์อื่นๆ ที่ไม่ใช่ซีพียู ซีพียูก็ต้องเสียเวลารอ ดังนั้นจึงเกิดแนวคิดการประมวลผลแบบให้หลายโปรแกรมทำงานพร้อมๆ กัน โดยมีการสลับช่วงการทำงานระหว่างโปรแกรมเพื่อสลับให้ซีพียูไปทำงานของทรานแซกชันอื่นๆ ซึ่งแต่ละทรานแซกชันที่ต้องการให้ซีพียูทำงานนั้นอาจจะเป็นโปรแกรมเดียวกันหรือต่างโปรแกรมก็ได้ โดยใช้หลักการอินเทอลีฟ มาใช้ในการควบคุมภาวะพร้อมกัน

อินเทอลีฟ (interleaved) คือ การที่ทรานแซกชันมากกว่าหนึ่งทรานแซกชัน มีการสลับการทำงานกันในขณะใดขณะหนึ่ง โดยที่ระบบจัดการฐานข้อมูลจะต้องควบคุมภาวะพร้อมกัน (concurrency control) เพื่อให้แต่ละทรานแซกชันมีการทำงานสลับกันไปมา ทั้งนี้ผลลัพธ์ที่ได้จะต้องมีความถูกต้องเสมือนว่าแต่ละทรานแซกชันทำงานเรียงลำดับทีละทรานแซกชันจนสิ้นสุดงานของแต่ละทรานแซกชันนั้น ตัวอย่างเช่น ภาพที่ 11.2 มีทราน

แซกชัน A และ B ทำงานพร้อมกันในช่วงเวลา time1 และ time2 โดยระบบจัดการฐานข้อมูลต้องควบคุมภาวะพร้อมกัน โดยมีการสลับการทำงานระหว่างทรานแซกชัน A และ B



ภาพที่ 11.2 การทำงานแบบอินเทอลีฟ เทียบกับ การประมวลผลในเวลาเดียวกัน

1.2 การประมวลผลในเวลาเดียวกัน (simultaneous processing) เป็นการทำงานในระบบคอมพิวเตอร์ที่มีซีพียูมากกว่า 1 ซีพียู เพื่อรองรับการทำงานของโปรแกรมใดโปรแกรมหนึ่งได้โดยไม่ต้องสลับทำงานระหว่างหลายทรานแซกชัน ดังนั้นซีพียูแต่ละตัวก็จะทำงานของโปรแกรมใดโปรแกรมหนึ่งแยกกันไปแต่ละซีพียูจนเสร็จงาน เช่น ภาพที่ 11.2 ทรานแซกชัน C และ D ทำงานในช่วงเวลา3 และ เวลา4 โดยแต่ละทรานแซกชันมีซีพียูประมวลผลแยกไปคนละซีพียู

ในการกล่าวถึงการควบคุมภาวะการทำงานพร้อมกันในที่นี้จะเป็นในสภานการณ์แบบอินเทอลีฟ โดยมีซีพียูที่จะทำงานเพียงซีพียูเดียวเท่านั้น

2. ปัญหาที่ทำให้มีการควบคุมภาวะพร้อมกัน

การควบคุมภาวะพร้อมกันในการใช้งานฐานข้อมูลเป็นสิ่งสำคัญอย่างยิ่ง เพราะหากระบบจัดการฐานข้อมูลไม่มีกลไกดังกล่าวย่อมจะก่อให้เกิดปัญหาในการทำงานดังนี้

2.1 ปัญหาการสูญหายของข้อมูลที่มีการปรับปรุงแก้ไข (the lost update problem) เป็นปัญหาที่เกิดจากทรานแซกชันมากกว่าหนึ่งทรานแซกชันต้องการปรับปรุงแก้ไขข้อมูล เดียวกันในเวลาไล่เรี่ยกัน ทำให้ผลลัพธ์ที่ได้ไม่ถูกต้อง เพราะข้อมูลที่ถูกแก้ไขโดยทรานแซกชันก่อนหน้าหายไปหมด จะปรากฏแต่ผลลัพธ์ที่เกิดจากการปรับปรุงแก้ไขของทรานแซกชันหลังสุดเท่านั้น

ตัวอย่างเช่น การฝากและถอนเงิน สมมติว่า จำนวนเงินที่มีในบัญชี ณ ปัจจุบันเท่ากับ 350 บาท โดยมีทรานแซกชันที่ 1 และ มีทรานแซกชันที่ 2 ต้องการฝากและถอนเงิน ซึ่งถ้าหากมีการทำงานตามลำดับก่อน-หลัง โดยให้ทรานแซกชันที่ 1 ทำการฝากเงินเท่ากับ 350 เข้าไปในฐานข้อมูลเสียก่อน หลังจากนั้นทรานแซกชันที่ 2

จะทำการถอนเงินในฐานข้อมูลก็ย่อมไม่มีปัญหาอะไร ดังตารางที่ 11.1 แสดงทำงานของทรานแซกชันที่ 1 และ 2 ในภาวะพร้อมกัน แบบเรียงลำดับก่อน-หลัง

ตารางที่ 11.1 การทำงานของทรานแซกชันในภาวะพร้อมกันแต่เป็นการเรียงลำดับก่อน-หลัง

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
ฝากเงิน 1000 (นั่นคือจำนวนเงินทั้งหมดในบัญชี 1000+350=1350)	Time 2		350
บันทึกจำนวนเงินทั้งหมด	Time 3		1350
	Time 4	อ่านจำนวนเงินในบัญชี	1350
	Time 5	ถอนเงินออกจากบัญชี 300 (นั่นคือ 1350-300=1050)	1350
	Time 6	บันทึกจำนวนเงินที่เหลือค้า	1050

ถ้าระบบจัดการฐานข้อมูลไม่มีกลไกในการควบคุมภาวะพร้อมกันจะเกิดปัญหาขึ้นได้ โดยหากทรานแซกชันที่ต้องการเรียกใช้ข้อมูลในการทำงานพร้อมกันโดยไม่เป็นลำดับก่อน-หลังทีละทรานแซกชัน ดังแสดงในตารางที่ 11.2 นั่นคือ สมมติว่าทรานแซกชันที่ 1 และทรานแซกชันที่ 2 เผอิญต้องการเรียกใช้ข้อมูลเดียวกันในเวลาไล่เลี่ยกัน ดังนี้

ตารางที่11.2 ปัญหาการสูญหายของข้อมูลที่มีการปรับปรุงแก้ไข

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
	Time 2	อ่านจำนวนเงินในบัญชี	350
ฝากเงิน 1000(นั้นคือจำนวนเงินทั้งหมดในบัญชี 1000+350=1350)	Time 3		350
	Time 4	ถอนเงินออกจากบัญชี 300(นั่นคือ350-300=50)	350
บันทึกจำนวนเงินทั้งหมด	Time 5		1350
	Time 6	บันทึกจำนวนเงินทั้งหมด	50

- ช่วงเวลา Time1 ทรานแซกชันที่ 1 อ่านจำนวนเงินในบัญชีมีค่าเท่ากับ 350 บาท

- ช่วงเวลา Time 2 ทรานแซกชันที่ 2 ก็อ่านอ่านจำนวนเงินในบัญชีมีค่าเท่ากับ 350 บาท เช่นกัน

- ช่วงเวลา Time 3 ทรานแซกชันที่ 1 ทำการฝากเงิน 1000 อีก 1000 บาท นั่นคือ 1000+350=1350 แต่ยังไม่ได้มีการบันทึกในฐานข้อมูล จึงทำให้ค่าข้อมูลในฐานข้อมูลเท่ากับ 350 บาทอยู่เช่นเดิม

- ช่วงเวลา Time 4 ทรานแซกชันที่ 2 ถอนเงินออกจากบัญชี 300 บาท (นั่นคือ 350-300=50) แต่ยังไม่ได้มีการบันทึกในฐานข้อมูล จึงทำให้ค่าในฐานข้อมูลเท่ากับ 350 บาทยอยู่เช่นเดิม

- ช่วงเวลา Time 5 ทรานแซกชันที่ 1 บันทึกค่าที่คำนวณได้เท่ากับ 1350 ลงในฐานข้อมูล

- ช่วงเวลา Time 6 ทรานแซกชันที่ 2 บันทึกค่าที่คำนวณเท่ากับ 50 ลงในฐานข้อมูล ซึ่งบันทึกทับข้อมูลเดิมที่มีการปรับปรุงแก้ไขโดยทรานแซกชันแรก ทำให้ข้อมูลสูญหายไปจากฐานข้อมูล โดยปรากฏแต่ข้อมูลที่มีการปรับปรุงแก้ไขโดยทรานแซกชันที่ 2 เท่านั้น

2.2ปัญหาจากการเรียกใช้ข้อมูลชุดเดียวกันของทรานแซกชันที่ยังไม่คอมมิต (uncommitted dependency problem) เป็นปัญหาที่เกิดจากทรานแซกชันมากกว่าหนึ่งทรานแซกชันต้องการเรียกใช้ข้อมูลชุดเดียวกัน โดยทรานแซกชันที่ 1 ยังอยู่ระหว่างกลางในการทำงาน ขณะเดียวกันทรานแซกชันที่ 2 เรียกใช้ข้อมูลที่แก้ไขโดยทรานแซกชันที่ 1 หลังจากนั้นปรากฏว่าทรานแซกชันที่ 1 มีปัญหา จะต้องถูกยกเลิกและโรลแบ็ก (rollback) เพื่อเริ่มต้นทำงานใหม่ทั้งหมด ดังนั้นข้อมูลที่ทรานแซกชันที่ 2 เรียกไปใช้งานไปแล้วจึงเป็นข้อมูลไม่ถูกต้อง ทำให้ผลลัพธ์ที่ได้ไม่ถูกต้องด้วย เพราะมีการยกเลิกไปแล้วจากทรานแซกชันที่ 1

ตัวอย่างเช่น ถ้าทรานแซกชันที่ 1 ต้องการฝากเงินอีก 1000 หน่วย ขณะที่ทรานแซกชันที่ 2 ต้องการถอนเงินออกไป 300 บาท ดังนั้น ถ้าหากทรานแซกชันที่ 1 และ 2 ทำงานเรียงลำดับโดยทรานแซกชันที่ 2 รอให้ทรานแซกชันที่ 1 ทำงานเสร็จเสียก่อน แล้วทรานแซกชันที่ 2 จึงเรียกใช้ข้อมูลเดียวกัน ปัญหาก็จะไม่เกิด และผลลัพธ์ของข้อมูลที่บันทึกในฐานข้อมูลก็จะถูกต้องด้วย ดังตารางที่ 11.3

ตารางที่ 11.3 การเรียกใช้ข้อมูลภายหลังจากมีการโรลแบ็ก

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
ฝากเงิน อีก 1000	Time 2		350
บันทึกจำนวนเงินทั้งหมด	Time 3		1350
มีปัญหาและการโรลแบ็ก	Time 4		350
	Time 5	อ่านจำนวนเงินในบัญชี	350
	Time 6	ถอนเงิน 30	350
	Time 7	บันทึกจำนวนเงินทั้งหมด	50

แต่ถ้าหากทรานแซกชันที่ 2 มีการเรียกใช้ข้อมูลที่ถูกปรับแก้โดยทรานแซกชันที่ 1 ซึ่งมีปัญหา และจะต้องถูกยกเลิกเพื่อเริ่มต้นทำงานนั้นใหม่ ก็จะทำให้ข้อมูลในฐานข้อมูลผิดพลาดไป ดังตารางที่ 11.4

ตารางที่ 11.4 การเรียกใช้ข้อมูลของทรานแซกชันที่ยังไม่คอมมิต

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
ฝากเงินอีก 1000	Time 2		350
บันทึกจำนวนเงินทั้งหมด	Time 3		1350
	Time 4	อ่านจำนวนเงินในบัญชี	1350
	Time 5	ถอนเงิน 300 (นั่นคือ 1350-300)	1350
มีปัญหาและต้องโรลแบ็ก	Time 6		350
	Time 7	บันทึกจำนวนเงินทั้งหมด	1050

2.3 ปัญหาการเรียกใช้ข้อมูลที่ไม่สอดคล้องกัน (inconsistent retrieval problem) เป็นปัญหาที่เกิดจากทรานแซกชันมากกว่าหนึ่งทรานแซกชัน มีการใช้งานชุดข้อมูลเดียวกัน โดยทรานแซกชันหนึ่งใช้ข้อมูลนั้นเพื่อประมวลผลใดๆ ในขณะเดียวกันก็มีทรานแซกชันอื่นได้มีการปรับปรุงแก้ไขข้อมูลชุดเดียวกัน ทำให้ผลลัพธ์ของทรานแซกชันแรกไม่ถูกต้อง

ตัวอย่าง ตารางที่ 11.5 แสดงคำสั่งภาษาเอสคิวแอลเพื่อสั่งให้ทรานแซกชันที่ 1 และที่ 2 ทำงาน ดังนี้

- ทรานแซกชันที่ 1 ต้องการทราบปริมาณสินค้าทั้งหมดโดยนำค่าปริมาณสินค้า (QPROD) แต่ละชนิดจากตารางสินค้า (product) มารวมกัน

- ทรานแซกชันที่ 2 ต้องการปรับปรุงแก้ไขยอดปริมาณสินค้าของสินค้า 2 ชนิด คือ รหัสสินค้า A3 เพิ่มอีก 30 หน่วย และรหัสสินค้า A4 หักออก 30 หน่วย

ตารางที่ 11.5 การทำงานของทั้งสองทรานแซกชันด้วยภาษาเอสคิวแอล

ทรานแซกชันที่ 1	ทรานแซกชันที่ 2
SELECT SUM (QPROD) FROM PRODUCT;	UPDATE PRODUCT SET QPROD = QPROD+30 WHERE PROD_ID = 'A3';
	UPDATE PRODUCT SET QPROD = QPROD-30 WHERE PROD_ID = 'A4';
	COMMIT;

ส่วนตารางที่ 11.6 แสดงข้อมูลในตารางสินค้าทั้งก่อนและหลังการทำงานของทรานแซกชันที่ 2 จากตารางที่ 11.5 ดังนี้

ตารางที่ 11.6 ข้อมูลในตารางสินค้า

	ปริมาณสินค้า (QPROD)	
รหัสสินค้า (PROD_ID)	ก่อนการทำงาน	หลังการทำงาน
A1	100	100
A2	120	120
A3	70	100 (เกิดจาก 70+30)
A4	35	5 (เกิดจาก 35-30)
A5	100	100
A6	30	30
Total	455	455

ถึงแม้ว่าผลลัพธ์ที่แสดงในตารางสินค้า (ตารางที่11.6) จะได้ผลลัพธ์ถูกต้อง แต่ถ้าหากดูการทำงานของแต่ละทรานแซกชัน ในแต่ละช่วงเวลาจะพบว่าผลรวมปริมาณสินค้าไม่ถูกต้อง ผลรวมที่ถูกต้องควรจะเป็น 455 แต่ผลรวมที่ได้จากการคำนวณ แสดงในตารางที่ 11.7 ได้เท่ากับ 485 ซึ่งยังไม่ถูกต้อง

ตารางที่ 11.7 ปัญหาการใช้ข้อมูลที่ไม่สอดคล้องกัน

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าข้อมูลในฐานข้อมูล	ผลรวมสินค้า
อ่านปริมาณสินค้าของ PROD_ID = “A1”	Time1		100	100
อ่านปริมาณสินค้าของ PROD_ID = “A2”	Time2		120	220
	Time 3	อ่านปริมาณสินค้าของ PROD_ID = “A3”	70	
	Time 4	ปริมาณสินค้า = 70+30		
	Time 5	บันทึกปริมาณสินค้าของ PROD_ID = “A3”	100	
อ่านปริมาณสินค้าของ PROD_ID = “A3”	Time 6		100	320
อ่านปริมาณสินค้าของ PROD_ID = “A4”	Time 7		35	355
	Time 8	อ่านปริมาณสินค้าของ PROD_ID = “A4”	35	
	Time 9	ปริมาณสินค้า = 35-30		
	Time 10	บันทึกปริมาณสินค้าของ PROD_ID = “A4”	5	
	Time 11	COMMIT		

อ่านปริมาณสินค้าของ PROD_ID = “A5”	Time 12		100	455
อ่านปริมาณสินค้า ของ PROD_ID = “A6”	Time 13		30	485

จากปัญหาที่เกิดขึ้นทั้ง 3 สถานการณ์ของการทำงานในภาวะพร้อมกัน ระบบจัดการฐานข้อมูลจึงต้องมีกลไกในการควบคุมภาวะพร้อมกัน เพื่อไม่ให้เกิดปัญหาที่กล่าวข้างต้น

ที่มา http://sot.swu.ac.th/Portals/156/sot/CP342/lesson11/ms2t1.htm