

create table SCB_bank
 (account char(3) not null,
 Acc_name char(15) not null,
 balance money not null)
 go
 create table TTB_bank
 (account char(3) not null,
 Acc_name char(15) not null,
 balance money not null)
 go

สร้างตารางของ ทั้ง 2 ธนาคารเพื่อมาทำการโอนเงิน
 ผมลืมแคปรูปเลยเอา code มาแปะ ไว้ และหลังจาก
 นั้น

ใส่ข้อมูล

insert SCB_bank values('001', 'Somchai',1500)
 insert SCB_bank values('002', 'Thitima',800)
 insert TTB_bank values('111','Somchai',400)
 insert TTB_bank values('112', 'Thitima',500)

Somchai (A)ต้องการโอนเงินจาก A → TMB_bank หมายเลข บัญชี 111 จำนวน 400 บาท

begin tran

update SCB_bank

set balance =balance -400

where account ='001'

เกิด exclusive lock ที่ ตาราง SCB_bank เพื่อแก้ไข

ขณะเดียวกัน Thitima (B)ต้องการโอนเงินจาก B → SCB_bank หมายเลข บัญชี 002 จำนวน 200 บาท

begin tran

update TMB_bank

set balance =balance -200

where account ='112'

เกิด exclusive lock ที่ ตาราง TMB_bank เพื่อแก้ไข

ถ้า Somchai (A) ในขั้นตอนที่ผ่านมาระบบเพิ่มยอดที่บัญชี ธนาคาร TMB_bank 400 บาท

update TMB_bank

set balance =balance + 400

where account ='111'

ทรานแซคชั่น ของ Somchai (A) พยายาม exclusive lock ที่ ตาราง TMB_bank เพื่อแก้ไข

แต่พบว่าไม่สามารถทำได้เนื่องจาก ทรานแซคชั่น ของ Thitima (B) ได้ทำการlock ไว้ก่อนหน้านี้แล้ว

ต้องรอนจนกว่าจะปลดล็อกออกมาเสียก่อน

กลับมาที่ทรานแซคชั่น Thitima (B) ทำการเพิ่มยอดในบัญชี SCB_bank จำนวน 200 บาท

update SCB_bank

set balance =balance + 200

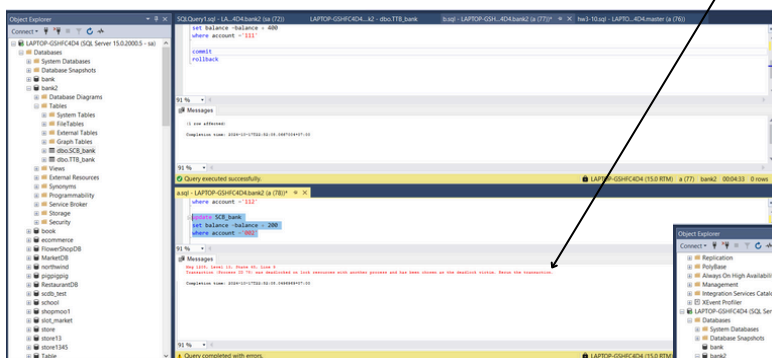
where account ='002'

ทรานแซคชั่น ของ Thitima (B) พยายาม exclusive lock ที่ ตาราง SCB_bank เพื่อแก้ไข

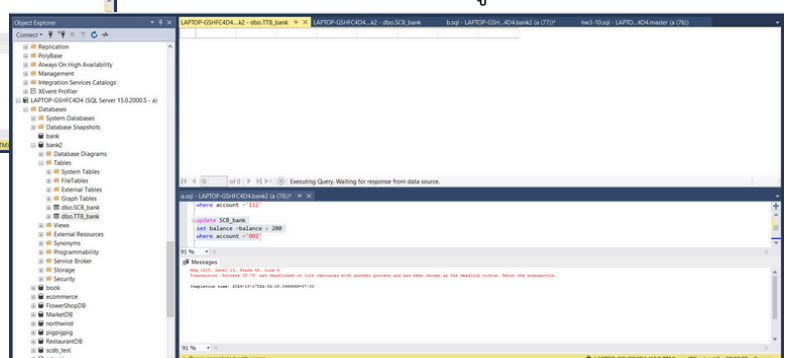
แต่พบว่าไม่สามารถทำได้เนื่องจาก ทรานแซคชั่น ของ Somchai (A)ได้ทำการ lock ไว้ก่อนหน้านี้แล้วต้องรอนจนกว่าจะปลด

ล็อกออกมาเสียก่อนขณะนี้เกิดการต่างคนต่างรอ จึงเกิด Deadlock

ตรวจสอบผลจากการเกิดDeadlock ในตารางของสองธนาคาร



ไม่สามารถดูตารางได้



การ recovery ข้อมูลที่คุณทำอยู่น่าจะเกี่ยวข้องกับการทำธุรกรรมในฐานข้อมูล SQL Server โดยใช้คำสั่ง BEGIN TRAN และ COMMIT TRAN เพื่อให้สามารถย้อนกลับ (rollback) หรือยืนยัน (commit) การทำธุรกรรมในกรณีที่เกิดปัญหา Transaction BEGIN TRAN ใช้คำสั่งนี้ ระบบจะเริ่มการทำงานเป็นธุรกรรม transaction ซึ่งการเปลี่ยนแปลงในฐานข้อมูล จะไม่ถูกบันทึกจนกว่าจะได้รับการยืนยัน COMMIT หรือยกเลิก ROLLBACK

การอัปเดตข้อมูลในฐานข้อมูล:

ในภาพกำลังอัปเดตยอดเงินใน SCB_bank และ TTB_bank โดยการลดเงินจากบัญชีหนึ่ง SCB_bank และเพิ่มเงินให้กับอีกบัญชีหนึ่ง TTB_bank ในรูปแบบที่ทำให้ยอดเงินยังคงสมดุล

ตัวอย่างในรูปที่แสดงคือการหักเงิน 400 จากบัญชีของ SCB account = '001' และเพิ่มเงิน 400 ไปที่บัญชี TTB account = '111'

การยืนยันธุรกรรม COMMIT เมื่อทำธุรกรรมเสร็จสิ้น คุณใช้คำสั่ง COMMIT เพื่อยืนยันว่าข้อมูลที่อัปเดตใน transaction นี้จะถูกบันทึกลงฐานข้อมูลจริง ๆ

แล้วก็จะสามารถเปิดดูตารางได้ปกติพร้อมกับตัวเลขที่อัปเดตไปแล้วเรียบร้อยแล้ว รูปข้างบน