หน่วยที่ 10 ความปลอดภัยของข้อมูล (Data Security)

เนื้อหา

ความปลอดภัยของระบบฐานข้อมูล เป็นการป้องกันผู้ไม่มีสิทธิเข้าใช้หรือแก้ไขข้อมูล การควบคุมความพร้อมกันในการเรียกใช้ข้อมูลเดียวกัน รวมถึงการรักษาความถูกต้องครบถ้วนสมบูรณ์ของข้อมูล โดยมีวัตถุประสงค์เพื่อ

1. รักษาความลับของข้อมูล (Secrecy)
2. ข้อมูลมีความถูกต้องสมบูรณ์ (Integrity)
3. มีฐานข้อมูลพร้อมใช้งานเสมอ (Availability)
4. ลดความเสี่ยง (Risk Assessment)

9.1 ความคงสภาพของข้อมูล (Data Integrity)

ความคงสภาพ (Integrity) หมายถึง ความถูกต้อง ความมั่นคง และความเป็นอันหนึ่งอันเดียวกัน ดังนั้น ความคงสภาพของข้อมูล จึงหมายความถึง ความถูกต้องของข้อมูลในฐานข้อมูล ฐานข้อมูลเชิงสัมพันธ์มีการกำหนดกฎเกณฑ์ของข้อมูลเพื่อรักษาความถูกต้องของข้อมูลและความสัมพันธ์ระหว่างข้อมูลต่าง ๆ และเป็นการป้องกันความเสียหายที่อาจเกิดกับฐานข้อมูล โดยมีจุดประสงค์หลัก คือ

(1) ป้องกันความผิดพลาดที่เกิดจากการเพิ่มข้อมูลลงในฐานข้อมูล
(2) รักษาความถูกต้องของข้อมูลเมื่อมีการเปลี่ยนแปลงข้อมูลในฐานข้อมูล
(3) ระบบจัดการฐานข้อมูลสามารถตัดสินใจได้ว่า จะจัดการกับข้อมูล ณ ตำแหน่งต่างๆ ในฐานข้อมูลอย่างไร

กฎเกณฑ์ในการคงความคงสภาพของข้อมูลสามารถแบ่งได้ 3 ประเภท ได้แก่

1. กฎเกณฑ์เกี่ยวกับชนิดของข้อมูล (Type constraint)

ข้อมูลแต่ละชนิดในฐานข้อมูลมีลักษณะการเก็บในฐานข้อมูลและการนำไปใช้งานต่างกัน เพื่อให้ข้อมูลเหล่านี้มีความถูกต้องก่อนการนำไปใช้งานจึงต้องมีการตรวจสอบชนิดและค่าของข้อมูลชนิดนั้นว่าถูกต้องหรือไม่

ตัวอย่าง 9.1 ชื่อนักศึกษา และ ชื่อธนาคาร อาจจะเป็นข้อมูลประเภทเดียวกันคือ ตัวอักษร แต่อาจมีข้อกำหนดที่แตกต่างกัน

ชื่อนักศึกษา	NOT NULL
ชื่อธนาคาร	IN (“ไทยพาณิชย์”, “กรุงไทย”, “กสิกรไทย”)

2. กฎเกณฑ์เกี่ยวกับฟิลด์ของข้อมูล (Attribute constraint)

กฎเกณฑ์ที่ใช้ในการกำหนดความถูกต้องของฟิลด์ข้อมูลสามารถแยกได้ 3 ประเภท ตามชนิดของแอททริบิว คือ

(1) ความคงสภาพของคีย์ (Key integrity) - การที่ค่าของคีย์จะต้องเป็นค่าที่มีเอกลักษณ์ ไม่ซ้ำกับข้อมูลใดในแถวอื่น

(2) ความคงสภาพของเอ็นทิตี้ (Entity integrity) - การที่ค่าของฟิลด์ที่เป็นคีย์หลักไม่สามารถเป็นค่าว่างได้

(3) ความคงสภาพของการอ้างอิงฟิลด์ (Referential integrity) - การอ้างอิงถึงฟิลด์จากความสัมพันธ์หนึ่งในความสัมพันธ์ใด จะต้องเป็นการอ้างอิงถึงฟิลด์ที่มีอยู่จริงในความสัมพันธ์นั้น การอ้างอิงถึงฟิลด์ที่ไม่มีอยู่จริงจะทำให้ไม่สามารถรักษาความคงสภาพของข้อมูลไว้ได้

ตัวอย่าง 9.2

พนักงาน (<u>รหัสพนักงาน,</u> ชื่อพนักงาน, ที่อยู่, รหัสแผนก)

<u>รหัสพนักงาน</u>	ชื่อพนักงาน	ทีอยู่	รหัสแผนก
C001	สมชาย	กรุงเทพฯ	01
C002	สมเกียรติ	นครสวรรค์	02
C001	จันจิรา	กรุงเทพฯ	03
NULL	น้ำฝน	กรุงเทพฯ	05

แผนก (<u>รหัสแผนก</u>, ชื่อแผนก)

<u>รหัสแผนก</u>	ชื่อแผนก
01	บุคคล
02	คอมพิวเตอร์
02	การตลาด
03	การเงิน

3. กฎเกณฑ์เกี่ยวกับฐานข้อมูล (Database constraint)

การประมวลผลที่เกิดขึ้นภายในฐานข้อมูล อันได้แก่ การอ่านข้อมูลและการเขียนข้อมูล มีผลโดยตรงต่อความคงสภาพของข้อมูล ผลลัพธ์จากการทำงานบางอย่างของฐานข้อมูลจะทำให้เกิดความเสียหายกับข้อมูลได้ หรืออาจจะทำให้ระบบล้ม ซึ่งจะมีผลให้ฐานข้อมูลอยู่ในสถานะที่ไม่มั่นคง (Unconsistency) ดังนั้นระบบจัดการฐานข้อมูลจึงจำเป็นต้องรักษากฎเกณฑ์ที่เกี่ยวกับการทำงานในฐานข้อมูลเอาไว้ เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นได้เหล่านี้

9.2 การฟื้นสภาพข้อมูล (Data Recovery)

การฟื้นสภาพ (recovery) คือ การที่ระบบจัดการฐานข้อมูลจัดการกับข้อมูลให้ย้อนไปอยู่ในสภาพเดิมที่ถูกต้อง ในการจัดการฐานข้อมูล การเกิดความขัดข้องหรือความเสียหายกับระบบไม่ว่ากรณีใด ๆ ที่อาจไปทำลายข้อมูลบางส่วน หรือทำให้ข้อมูลนั้นไม่ถูกต้อง ไม่น่าเชื่อถือได้ ดังนั้นจึงต้องมีวิธีการเพื่อจะนำข้อมูลที่ถูกทำลายไปกลับคืนมาและอยู่ในสภาพที่ถูกต้องน่าเชื่อถือดังเดิม และเป็นการทำให้มั่นใจความขัดข้องต่างๆ จะไม่ก่อให้เกิดความเสียหายกับฐานข้อมูล

วิธีการฟื้นสภาพ

การแก้ปัญหาความขัดข้องต่างๆ จะใช้ ไฟล์ประวัติ (log file) เข้ามาช่วยในการฟื้นสภาพ ด้วยการบันทึกรายการต่าง ๆ ลงในไฟล์ประวัติ ซึ่งเรียกวิธีนี้ว่า การฟื้นสภาพแบบล็อกเบส (log-based recovery) ข้อมูลในไฟล์ประวัติจะบอกถึงสถานะของข้อมูลและสถานะของรายการ โดยจะมีรายละเอียดประกอบด้วย

- หมายเลขรายการ (Transaction ID)
- ชื่อข้อมูลที่ถูกบันทึก (Data Item Name)
- ค่าเก่า (old value) คือ ค่าของข้อมูลก่อนการบันทึก
- ค่าใหม่ (new value) คือ ค่าของข้อมูลหลังการบันทึก

รูปแบบของคำสั่งที่จัดเก็บในไฟล์ประวัติ ประกอบด้วย

1. เริ่มต้นรายการ

<T starts>

T = หมายเลขรายการ

2. แก้ไขข้อมูล

<T, X, old: V, new: W>

T = หมายเลขรายการ X = ชื่อข้อมูล V = ค่าเก่า W = ค่าใหม่

3. เสร็จสิ้นการทำงาน

<T commits>

T = หมายเลขรายการ

แบบฝึกหัด 9.1

กำหนดให้การทำงานเริ่มต้นด้วยรายการ 1 (T1) และตามด้วยรายการ 2 (T2)

T1:		T2:	
	Read(A)		Read(A)
	A ← A + 50		A ← A + 10
	Read(B)		Write(A)
	B ← B + 100		Read(D)
	Write(B)		D ← D – 10
	Read(C)		Read(E)
	C ← 2C		Read(B)
	Write(C)		E ← E + B
	A ← A + B + C		Write(E)
	Write(A)		D ← D + E
			Write(D)

ถ้าค่าเริ่มต้นของ A = 100, B = 300, C = 5, D = 60, E = 80 เขียนคำสั่งในไฟล์ประวัติสำหรับรายการข้างต้นทั้งสองรายการ

9.3 การควบคุมภาวะความพร้อมกัน (Concurrency Control)

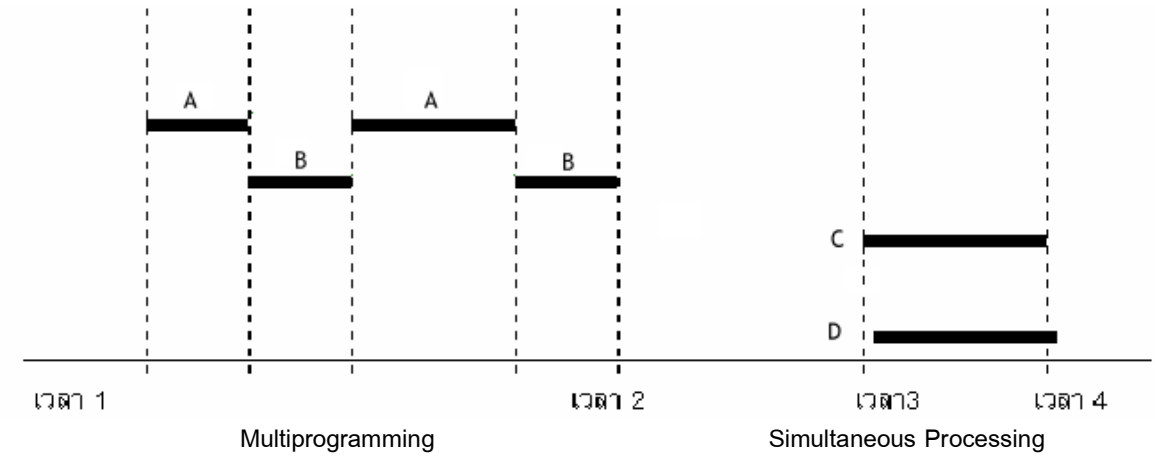
การที่มีรายการหลายๆ รายการต้องการเรียกใช้ข้อมูลเดียวกันในเวลาเดียวกันจากฐานข้อมูลเพื่อทำงานของแต่ละรายการ ภาวะการทำงานพร้อมกันเกิดจากระบบการทำงานได้ 2 ระบบ

1. การทำงานในระบบหลายโปรแกรม หรือการทำงานแบบมัลติโปรแกรมมิง (multiprogramming) เป็นการทำงานของระบบคอมพิวเตอร์ที่ออกแบบเพื่อให้หน่วยประมวลผลผลกลางหรือซีพียู (Central Processing Unit; CPU) ทำงานหลายๆ งานในขณะเดียวกันได้ โดยใช้วิธีการสลับช่วงการทำงานระหว่างโปรแกรมเพื่อสลับให้ซีพียูไปทำงานของรายการอื่นๆ ซึ่งแต่ละรายการที่ต้องการให้ซีพียูทำงานนั้นอาจจะเป็นโปรแกรมเดียวกันหรือต่างโปรแกรมก็ได้ โดยใช้หลักการอินเทอลีฟ มาใช้ในการควบคุมภาวะพร้อมกัน

อินเทอลีฟ (interleaved) คือ การที่รายการมากกว่าหนึ่งรายการ มีการสลับการทำงานกันในขณะใดขณะหนึ่ง โดยที่ระบบจัดการฐานข้อมูลจะต้องควบคุมภาวะพร้อมกัน (concurrency control) เพื่อให้แต่ละรายการมีการทำงานสลับกันไปมา ทั้งนี้ผลลัพธ์ที่ได้จะต้องมีความถูกต้องเสมือนว่าแต่ละรายการทำงานเรียงลำดับทีละรายการจนสิ้นสุดงานของแต่ละรายการนั้น

2. การประมวลผลในเวลาเดียวกัน (simultaneous processing) เป็นการทำงานในระบบคอมพิวเตอร์ที่มีซีพียูมากกว่า 1 ซีพียู เพื่อรองรับการทำงานของโปรแกรมใดโปรแกรมหนึ่งได้โดยไม่ต้องสลับทำงานระหว่างหลายรายการ ดังนั้นซีพียูแต่ละตัวก็จะทำงานของโปรแกรมใดโปรแกรมหนึ่งแยกกันไปแต่ละซีพียูจนเสร็จงาน

ตัวอย่าง 9.3



วิธีการควบคุมภาวะความพร้อมกัน

ใช้หลักการจัดลำดับความขัดแย้ง (Conflict Serializability) ด้วยการพิจารณาคำสั่งในการทำงานที่ขัดแย้งกัน และจัดลำดับคำสั่งใหม่เพื่อหลีกเลี่ยงความขัดแย้ง

กำหนดรายการ T1 & T2 สามารถพิจารณาความขัดแย้งได้ ดังนี้

1. Write/Read Conflict

T1	T2
write X	
	read X

T2 อ่านค่าภายหลังจากที่ T1 เขียนข้อมูล ซึ่งเป็นข้อมูลที่เปลี่ยนแปลงไปแล้ว

2. Read/Write Conflict

T1	T2
read X	
	write X

T1 อ่านข้อมูลก่อนที่ T2 จะเขียนข้อมูล ซึ่งเป็นข้อมูลเก่าก่อนการเปลี่ยนแปลง

3. Write/Write Conflict

T1	T2
write X	
	write X

T2 เขียนค่าทับค่าเดิมที่ T1 เขียน ทำให้ข้อมูลสูญหาย

4. No Conflict

T1	T2
read X	
	read X

T1 และ T2 อ่านข้อมูลเดียวกัน ทำให้ไม่เกิดความขัดแย้ง

ปัญหาที่ทำให้มีการควบคุมภาวะความพร้อมกัน

การควบคุมภาวะพร้อมกันในการใช้งานฐานข้อมูลเป็นสิ่งสำคัญอย่างยิ่ง เพราะหากระบบจัดการฐานข้อมูลไม่มีกลไกดังกล่าวย่อมจะก่อให้เกิดปัญหาในการทำงานดังนี้

1. ปัญหาการสูญหายของข้อมูลที่มีการปรับปรุงแก้ไข (the lost update problem)

ปัญหาที่เกิดจากรายการมากกว่าหนึ่งรายการต้องการปรับปรุงแก้ไขข้อมูล เดียวกันในเวลาไล่เรี่ยกัน ทำให้ผลลัพธ์ที่ได้ไม่ถูกต้อง เพราะข้อมูลที่ถูกแก้ไขโดยรายการก่อนหน้าหายไปหมด จะปรากฏแต่ผลลัพธ์ที่เกิดจากการปรับปรุงแก้ไขของรายการหลังสุดเท่านั้น

แบบฝึกหัด 9.2

การฝากและถอนเงิน จำนวนเงินที่มีในบัญชี ณ ปัจจุบันเท่ากับ 350 บาท โดยมีรายการที่ 1 และ มีรายการที่ 2 ต้องการฝากและถอนเงิน

การเรียกใช้ข้อมูลในการทำงานพร้อมกันโดยไม่เป็นลำดับก่อน-หลังทีละรายการ

รายการที่ 1	เวลา	รายการที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
	Time 2	อ่านจำนวนเงินในบัญชี	350
ฝากเงิน 1000	Time 3		350
	Time 4	ถอนเงินออกจากบัญชี 300	350
บันทึกจำนวนเงินทั้งหมด	Time 5		1350
	Time 6	บันทึกจำนวนเงินทั้งหมด	50

จัดลำดับการทำงานของรายการด้วยการใช้วิธีการจัดลำดับความขัดแย้ง

รายการที่ 1	เวลา	รายการที่ 2	ค่าของข้อมูลในฐานข้อมูล
	Time 1		
	Time 2		
	Time 3		
	Time 4		
	Time 5		
	Time 6		

2. ปัญหาจากการเรียกใช้ข้อมูลชุดเดียวกันของรายการที่ยังไม่คอมมิต (uncommitted dependency problem)

ปัญหาที่เกิดจากรายการมากกว่าหนึ่งรายการต้องการเรียกใช้ข้อมุลชุดเดียวกัน โดยรายการที่ 1 ยังอยู่ระหว่างกลางในการทำงาน ขณะเดียวกันรายการที่ 2 เรียกใช้ข้อมูลที่แก้ไขโดยรายการที่ 1 หลังจากนั้นปรากฏว่ารายการที่ 1 มีปัญหา จะต้องถูกยกเลิกและเริ่มต้นทำงานใหม่ทั้งหมด ดังนั้นข้อมูลที่รายการที่ 2 เรียกไปใช้งานไปแล้วจึงเป็นข้อมูลไม่ถูกต้อง ทำให้ผลลัพธ์ที่ได้ไม่ถูกต้องด้วย เพราะมีการยกเลิกไปแล้วจากรายการที่ 1

แบบฝึกหัด 9.3

รายการที่ 1 ต้องการฝากเงิน 1000 บาท ขณะที่รายการที่ 2 ต้องการถอนเงิน 300 บาท รายการที่ 2 มีการเรียกใช้ข้อมูลที่ถูกปรับแก้โดยรายการที่ 1 ซึ่งมีปัญหา และจะต้องถูกยกเลิกเพื่อเริ่มต้นทำงานนั้นใหม่

รายการที่ 1	เวลา	รายการที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนเงินในบัญชี	Time 1		350
ฝากเงิน 1000	Time 2		350
บันทึกจำนวนเงินทั้งหมด	Time 3		1350
	Time 4	อ่านจำนวนเงินในบัญชี	1350
	Time 5	ถอนเงิน 300	1350
มีปัญหาและต้องเริ่มต้นใหม่	Time 6		350
	Time 7	บันทึกจำนวนเงินทั้งหมด	1050

จัดลำดับการทำงานของรายการด้วยการใช้วิธีการจัดลำดับความขัดแย้ง

รายการที่ 1	เวลา	รายการที่ 2	ค่าของข้อมูลในฐานข้อมูล
	Time 1		
	Time 2		
	Time 3		
	Time 4		
	Time 5		
	Time 6		
	Time 7		

3. ปัญหาการเรียกใช้ข้อมูลที่ไม่สอดคล้องกัน (inconsistent retrieval problem)

ปัญหาที่เกิดจากรายการมากกว่าหนึ่งรายการ มีการใช้งานชุดข้อมูลเดียวกัน โดยรายการหนึ่งใช้ข้อมูลนั้นเพื่อประมวลผลใดๆ ในขณะเดียวกันก็มีรายการอื่นได้มีการปรับปรุงแก้ไขข้อมูลชุดเดียวกัน ทำให้ผลลัพธ์ของรายการแรกไม่ถูกต้อง

แบบฝึกหัด 9.4

รายการที่ 1 ต้องการทราบปริมาณสินค้าทั้งหมดโดยนำค่าปริมาณสินค้า (Quan) แต่ละชนิดจากตารางสินค้า (Product) มารวมกัน รายการที่ 2 ต้องการปรับปรุงแก้ไขยอดปริมาณสินค้าของสินค้า 2 ชนิด คือ รหัสสินค้า A3 เพิ่มอีก 30 หน่วย และรหัสสินค้า A4 หักออก 20 หน่วย ได้คำสั่ง SQL ดังนี้

T1	T2
SELECT Sum(Quan) FROM Product;	UPDATE Product SET Quan = Quan + 30 WHERE Prod_ID = 'A3';
	UPDATE Product SET Quan = Quan - 20 WHERE Prod_ID = 'A4';

ข้อมูลในตารางสินค้าทั้งก่อนและหลังการทำงานของรายการที่ 2

รหัสสินค้า (Prod_ID)	ปริมาณสินค้า (Quan) ก่อนการทำงาน	ปริมาณสินค้า (Quan) หลังการทำงาน
A1	100	100
A2	120	120
A3	70	100
A4	35	15
A5	100	100
A6	30	30
รวม	455	465

จัดลำดับการทำงานของรายการด้วยการใช้วิธีการจัดลำดับความขัดแย้ง

รายการที่ 1	เวลา	รายการที่ 2	ค่าข้อมูลในฐานข้อมูล	ผลรวมสินค้า
	Time1			
	Time2			
	Time 3			
	Time 4			
	Time 5			
	Time 6			
	Time 7			
	Time 8			
	Time 9			
	Time 10			
	Time 11			
	Time 12			
	Time 13			

9.4 การล็อก (Locking)

จากปัญหาที่เกิดขึ้นทั้ง 3 กรณีของการทำงานในภาวะพร้อมกัน ระบบการจัดการฐานข้อมูลจึงต้องมีกลไกในการควบคุมภาวะพร้อมกัน เพื่อไม่ให้เกิดปัญหาข้างต้น

ประเภทของการล็อก (Type of Locking)

1. การล็อกแบบผูกขาด (exclusive locks) การกำหนดสถานะของข้อมูลให้เป็นล็อกโดยไม่ให้รายการอื่นใช้ข้อมูลที่ถูกล็อกนั้น เป็นการล็อกที่ผู้ใช้ล็อกแบบนี้จะเป็นผู้เดียวที่สามารถเปลี่ยนแปลงข้อมูลที่ล็อกไว้ได้ผู้เดียว ผู้ใช้คนอื่น ๆ จะไม่สามารถอ่านหรือแก้ไขระเบียนนั้นได้เลย จนกว่าจะมีการปล่อยล็อก
2. การล็อกแบบแบ่งส่วน (shared locks) การกำหนดสถานะของข้อมูลให้ใช้งานร่วมกันกับรายการอื่นได้

ระดับการล็อก (Locking Level)

ในการควบคุมภาวะการเข้าถึงข้อมูลพร้อมกัน โดยการใช้การล็อกนี้จะสามารถเลือกรูปแบบของการล็อกข้อมูลได้หลายระดับ ตามลักษณะของงานนั้น ซึ่งสามารถแบ่งระดับของการล็อกนี้เป็นระดับต่าง ๆ ดังนี้

(1) การล็อกฐานข้อมูล (entire database) เป็นการล็อกทั้งฐานข้อมูลขณะที่รายการใดรายการหนึ่งกำลังใช้งาน การล็อกแบบนี้จะล็อกในระหว่างที่มีการสำรอง (Backup) ฐานข้อมูลทั้งหมดนั้นขึ้นเทป

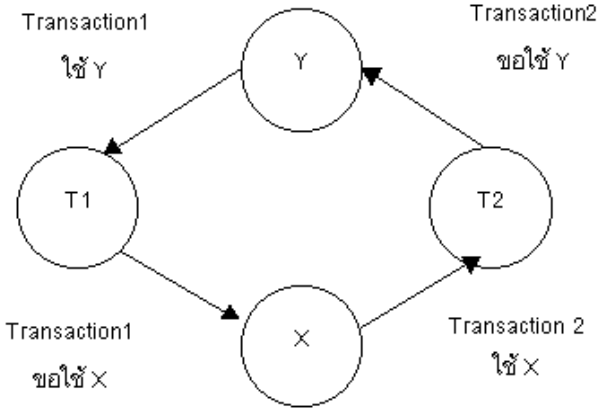
(2) การล็อกเฉพาะตารางใดตารางหนึ่งในฐานข้อมูล (a relation) การล็อกตารางเป็นการล็อกตารางข้อมูลที่ต้องการซึ่งระเบียนต่างๆ ที่อยู่ในตารางนั้นจะถูกล็อกด้วยโดยอัตโนมัติ ผู้ใช้คนอื่นจะไม่สามารถใช้ข้อมูลใดที่อยู่ในตารางที่ถูกล็อกนี้ได้เลย จนกว่าจะมีการปล่อยล็อก การล็อกตารางจะถูกทำในกรณีที่จะมีการปรับปรุงหรือแก้ไขข้อมูลทั้งหมดในตารางนั้น ๆ

(3) การล็อกเฉพาะแถวบางแถว (a tuple) ในฐานข้อมูล เป็นการล็อกเฉพาะระเบียนหรือแถวข้อมูลใด ๆ ในตาราง ซึ่งผู้ใช้คนอื่นจะไม่สามารถแก้ไขหรือปรับปรุงระเบียนที่ถูกล็อกนี้ได้จนกว่าจะมีการปล่อยล็อก การล็อกระเบียนจะถูกใช้มากในการเขียนโปรแกรมที่ใช้ในการปรับปรุงระเบียนใดระเบียนหนึ่งในตาราง การล็อกระเบียนเป็นการห้ามไม่ให้ผู้อื่นเข้าถึงข้อมูลที่ถูกล็อกนี้ได้ ไม่ว่าจะเป็นการดูข้อมูลหรือการแก้ไขข้อมูล

(4) การล็อกเฉพาะบางฟิลด์ (field) หรือบางแอตทริบิวต์ (attribute) ในฐานข้อมูลเท่านั้น การล็อกเฉพาะบางฟิลด์เป็นการล็อกเฉพาะเขตข้อมูลใดๆในระเบียนที่ต้องการของตาราง การล็อกในระดับนี้จะใช้ในกรณีที่มีการแก้ไขข้อมูลในเขตข้อมูลที่ทำการล็อกนั้นบ่อย ๆ

เดดล็อก (dead lock)

เหตุการณ์ที่รายการรอการใช้ข้อมูลที่ถูกล็อกโดยรายการอื่นอย่างไม่รู้จบในลักษณะเป็นลูกโซ่



วิธีการแก้ปัญหาการเกิดเดดล็อก มีดังนี้

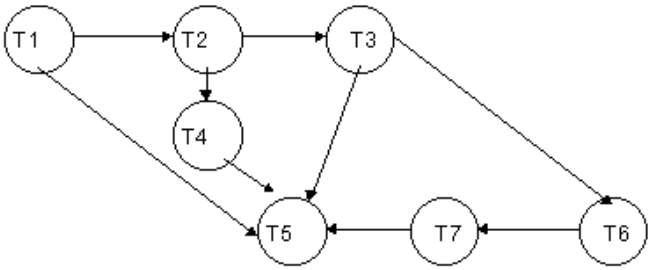
(1) การป้องกันการเกิดเดดล็อก (deadlock prevention) - ระบบจัดการฐานข้อมูลในส่วนของการควบคุมภาวะพร้อมกันจะกำหนดว่า รายการที่ต้องการเรียกใช้ข้อมูลใดก็ตามจะต้องล็อกข้อมูลทุกอันที่ต้องการเรียกใช้ทั้งหมดไว้ก่อนการใช้งาน ถ้าหากรายการไม่สามารถจะล็อกข้อมูลไว้ล่วงหน้าได้ จะต้องรอจนกว่าจะล็อกได้ครบเสียก่อนจึงจะเริ่มต้นทำงานได้

(2) การตรวจจับการเกิดเดดล็อก (deadlock detection) - ระบบจัดการฐานข้อมูลจะตรวจจับว่ารายการใดทำให้เกิดเดดล็อกบ้าง โดยจะตรวจจับจากกราฟที่เรียกว่า “wait-for graph” หลักการโดยทั่วไปคือ ตรวจหาการรอเป็นวัฏจักร (Circular wait) ระหว่างรายการต่าง ๆ โดยเครื่องมือที่สำคัญ คือ resource allocation graph หรือ

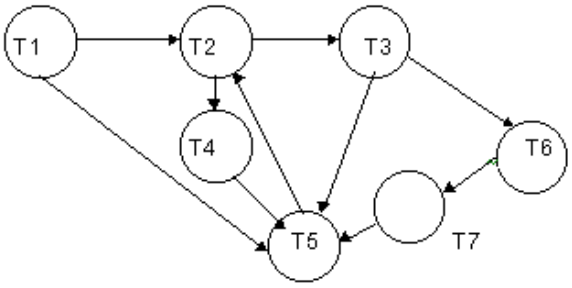
wait-for graph ซึ่งเป็นกราฟแสดงความสัมพันธ์ระหว่างทรัพยากรและกระบวนการต่าง ๆ ในแง่ของการจับจองใช้และการขอใช้

T1 รอข้อมูลที่ถูกล็อกโดย T2 และT5
T2 รอข้อมูลที่ถูกล็อกโดย T3และ T4
T3 รอข้อมูลที่ถูกล็อกโดย T5 และT6
T4 รอข้อมูลที่ถูกล็อกโดย T5
T6 รอข้อมูลที่ถูกล็อกโดยT7
T7 รอข้อมูลที่ถูกล็อกโดยT5

ระบบจัดการฐานข้อมูลจะใช้ข้อมูลจากตัวอย่างข้างต้นสร้างเป็นกราฟเพื่อตรวจจับเดดล็อก



ถ้ามีการเพิ่มเติมโดยให้รายการที่ 5 รอข้อมูลที่ถูกล็อกโดยรายการที่ 2 จะทำให้เกิดเดดล็อก



ถ้าหากพบว่ามีรายการใน wait for graph นั่นคือ มีการรอในลักษณะวนเป็นลูกโซ่ ซึ่งเป็น wait for graph ที่เกิดเดดล็อก ระบบจัดการฐานข้อมูลจะตรวจสอบและยกเลิกการทำงานของรายการที่ทำให้เกิดเดดล็อก ซึ่งอาจจะมีการเริ่มต้นทำงานใหม่ทั้งหมด

9.5 ความปลอดภัย (Security)

สิ่งสำคัญในการสร้างระบบรักษาความปลอดภัยในระบบฐานข้อมูลก็คือการกำหนดผู้ใช้งานในระบบฐานข้อมูล นั่นคือ การที่ผู้ใดจะเข้ามาใช้ระบบฐานข้อมูลได้จะต้องได้รับการอนุญาตก่อน นอกจากนี้เมื่อเข้าระบบได้แล้ว ผู้ใช้งานนั้นสามารถทำอะไรได้บ้างต้องขึ้นอยู่กับการให้สิทธิของผู้บริหารฐานข้อมูล

การสร้างสิทธิผู้ใช้ในการเข้าถึงข้อมูล

(1) การยืนยันตัวบุคคล (Authentication) เพื่อให้มั่นใจได้ว่าผู้ที่จะเข้าระบบเป็นผู้ที่มีสิทธิจริง ในปัจจุบันนี้มีการใช้เทคนิคมากมายในการยืนยันตัวบุคคล แต่ที่เป็นที่นิยมได้แก่

- การใช้รหัสผ่าน(password)
- การใช้บัตรสมาร์ดคาร์ด (smartcard)
- การใช้การตรวจสอบจากร่างกายมนุษย์ (biometric)

(2) การให้สิทธิ (Authorization) ผู้ใช้งานระบบฐานข้อมูลมีสิทธิในการใช้ข้อมูลแตกต่างกันมากมาย เช่น

- สิทธิในการอ่านข้อมูลหรือเรียกดูข้อมูล (read)
- สิทธิในการเพิ่มข้อมูล (insert)
- สิทธิในการเปลี่ยนแปลงข้อมูล (update)
- สิทธิในการลบข้อมูล (delete)
- สิทธิในการสร้างดัชนี (index)
- สิทธิในการสร้างตารางหรือวิว (resource)
- สิทธิในการเปลี่ยนแปลงโครงสร้างข้อมูล (alteration)
- สิทธิในการลบตารางหรือวิว (drop)

การสร้างข้อมูลให้เป็นความลับ

(1) การเข้ารหัส (coding) เป็นกระบวนการแปลงรูปแบบของข้อมูลให้อยู่ในรูปที่บุคคลอื่นๆ ไม่สามารถรู้เนื้อหาของข้อมูล ยกเว้นบุคคลที่เป็นผู้รับ ซึ่งจะต้องมีตัวถอดรหัสทำการแปลงข้อมูลนั้นกลับมาเป็นข้อมูลต้นฉบับ การเข้ารหัสจะใช้วิธีแทนค่าแต่ละค่าด้วยค่าอื่น ซึ่งเป็นการป้องกันข้อมูลในระดับหนึ่ง สามารถป้องกันผู้ที่ไม่ทราบวิธีการเข้ารหัสใช้ข้อมูลได้อย่างง่ายๆ

(2) การบีบอัดข้อมูล (compression) มักจะใช้กับข้อมูลประเภทตัวเลข หรือข้อมูลที่แปลงเป็นเลขฐานสองแล้ว

(3) การแทนค่า (substitution) มีหลักการทำงานคล้ายกับการเข้ารหัสโดยมีการกำหนดค่าที่จะแทนไว้ล่วงหน้า ส่วนการเข้ารหัสจะเป็นการกำหนดหลักการเข้ารหัสไว้

(4) การสลับตำแหน่งข้อมูล (transposition) ทำโดยไม่ได้เปลี่ยนข้อมูล แต่ใช้วิธีการสลับตำแหน่งของข้อมูลแทนในการใช้งานจริงในการรักษาความปลอดภัยของฐานข้อมูลมักจะเป็นการนำเทคนิคต่างๆ หลายเทคนิคมาประยุกต์ใช้งานร่วมกัน เพื่อให้ระบบความปลอดภัยนั้นมั่นคงและเชื่อถือได้

9.6 การกำหนดสิทธิการใช้งาน

การกำหนดสิทธิในการเข้าถึงข้อมูล และมอบอำนาจการเข้าถึงข้อมูลตลอดจนเรียกคืนอำนาจได้ สามารถระบุสิทธิผู้ใช้ในระบบด้วยภาษา SQL

1. การสร้างรหัสให้แก่ผู้ใช้

การกำหนดรหัสผ่านให้แก่ผู้ใช้ โดยใช้คำสั่ง CREATE

รูปแบบ CREATE <ชื่อผู้ใช้>
IDENTIFIED BY <พาสเวิร์ด>;

2. การกำหนดสิทธิการเข้าถึงข้อมูล

การกำหนดสิทธิการเข้าถึงข้อมูลเป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคนมีสิทธิกระทำการใดกับข้อมูลในตารางใดได้บ้างหรือการกำหนดให้มีสิทธิดูข้อมูลได้เพียงอย่างเดียว

(1) การกำหนดสิทธิในการเข้าถึงข้อมูล ได้แก่ การเรียกค้นข้อมูลด้วยคำสั่ง (SELECT) การเพิ่มข้อมูลด้วยคำสั่ง (INSERT) การ ลบข้อมูลมูลด้วยคำสั่ง (DELETE) หรือการปรับปรุง มูลด้วยคำสั่ง (UPDATE)

รูปแบบ GRANT <SELECT,INSERT,UPDATE,DELETE> ON <ชื่อตาราง> TO <ชื่อผู้ใช้>;

(2) การให้สิทธิในการเข้าถึงข้อมูลทั้งหมด

ใช้ ALL PRIVILEGES (หรือ ALL เท่านั้น) ในคำสั่ง GRANT

รูปแบบ GRANT ALL ON <ชื่อตาราง> TO <ชื่อผู้ใช้>;

(3) การให้สิทธิในการเรียกดูข้อมูลแก่ผู้ใช้ทุกคน

ใช้ PUBLIC กับคำสั่ง SELECT ควบคู่ไปกับคำสั่ง GRANT

รูปแบบ GRANT SELECT ON <ชื่อตาราง> TO PUBLIC;

แบบฝึกหัด 9.5

1. สร้างผู้ใช้ชื่อ Wichai รหัสผ่าน BENZ2000

2. กำหนดสิทธิในการเรียกดูข้อมูลให้ Wichai มีสิทธิเรียกดูข้อมูลในตาราง CUSTOMERS

3. กำหนดสิทธิให้ Wichai และ Thidarat สามารถเรียกดูข้อมูล และเพิ่มข้อมูลได้ในตาราง ORDERS

4. กำหนดสิทธิให้ Thidarat มีสิทธิเปลี่ยนค่าในคอลัมน์ SALECOM ในตาราง SALES

5. กำหนดให้ Nattapol สามารถทำคำสั่งใด ๆ ในตาราง CUSTOMERS ได้

6. กำหนดให้ผู้ใช้คนใดก็ได้เข้าไปดูตาราง ORDERS

3. <u>การยกเลิกสิทธิการเข้าถึงข้อมูล</u>

การยกเลิกสิทธิใดๆแก่ผู้ใช้ตามที่ได้ใช้กำหนดสิทธิการเข้าถึงข้อมูลไว้

รูปแบบ REVOKE <SELECT,INSERT,UPDATE,DELETE>ON <ชื่อตาราง> FROM <ชื่อผู้ใช้>;

แบบฝึกหัด 9.6

1. ยกเลิกสิทธิในการเรียกดูข้อมูลในตาราง CUSTOMERS ของ Wichai

2. ยกเลิกสิทธิในการแก้ไขข้อมูลในตาราง SALES ของ Thidarat

3. ยกเลิกสิทธิในการเพิ่มเติมข้อมูลในตาราง ORDERS ของ Thidarat

9.6 วิว(View)

การควบคุมความปลอดภัยให้กับข้อมูลสามารถสร้างโครงสร้างข้อมูลใหม่ ที่ทำให้ผู้ใช้ เห็นเพียงโครงสร้างบางส่วนของฐานข้อมูลเท่านั้น ที่เป็นการป้องกันไม่ให้ผู้ใช้ได้เห็นข้อมูลทั้งหมดของฐานข้อมูล เรียกว่า “ตารางเสมือน” หรือ “วิว”

วิวเป็นตารางข้อมูลที่มีรายละเอียดหรือได้รายละเอียดมาจากตารางหลัก ถูกสร้างขึ้นจากฐานข้อมูล โดยตารางที่สร้างขึ้นนี้จะสอดคล้องกับการใช้งานของผู้ใช้และยังเป็นการป้องกันข้อมูลที่แท้จริงภายในฐานข้อมูล และเป็นกลไกรักษาความปลอดภัยในการปกปิดส่วนต่าง ๆ ของตารางที่เป็นความลับหรือเกินความจำเป็นสำหรับผู้ใช้

1. การสร้างวิว

การสร้างวิวอาจสร้างมาจากตารางข้อมูลเดียวหรือมากกว่าหนึ่งตารางได้ โดยใช้คำสั่ง CREATE VIEW

รูปแบบ CREATE VIEW <ชื่อวิว>
AS <คำสั่ง SELECTt>;

แบบฝึกหัด 9.7 ตาราง SALES

SALENO	SALENAME	ADDRESS	SALECOM
1001	Chaiwat	Bangkok	0.12
1002	Mitree	Puket	0.13
1004	Benjawan	Bangkok	0.11
1007	Kanjana	Chiangmai	0.15
1003	Ternjai	Nonthaburi	0.10

1. สร้างวิวชื่อ SALESOWN จากตาราง SALES โดยให้มีคอลัมน์ SALENO และ SALENAME

2. สร้างวิวชื่อ BANGKOKSTAFF จากตาราง SALES โดยให้มีคอลัมน์ ADDRESS ที่เป็น “Bangkok”

2. การสอบถามข้อมูลจากวิว

ใช้คำสั่ง SELECT

รูปแบบ SELECT <ชื่อฟิลด์>
FROM <ชื่อวิว>
[WHERE <ชื่อฟิลด์> = <ค่า>;

3. การลบโครงสร้างของวิว

รูปแบบ DROP VIEW <ชื่อวิว>;

แบบฝึกหัด 9.8

1. เรียกดูข้อมูลทั้งหมดจากวิว BANGKOKSTAFF

2. ลบโครงสร้างของวิวชื่อ BANGKOKSTAFF

แบบฝึกหัด 9.9 ตาราง Employee

EMP_ID	NAME	SURNAME	SEX	SALARY	DEPT_ID
1001	ธนา	ดีเลิศ	M	15,000	003
1002	ศิรินุช	รักเรียน	F	18,000	001
1003	ศิริชัย	พลาสัย	M	16,000	003
1004	ฉวีวรรณ	งานดี	F	18,000	002
1005	ประนอม	ร่าเริง	F	18,000	001

1. สร้างวิว Hemployee สำหรับพนักงานที่มีเงินเดือนน้อยกว่า 18,000

2. สร้างวิว Vemployee ที่แสดง EMP_ID, NAME และ SURNAME

3. สร้างวิว Gemployee ในการสรุปจำนวนของพนักงานที่มีเงินเดือนที่เท่ากัน

ตาราง Department

DEPT_ID	DEPTNAME
001	ฝ่ายลูกค้าสัมพันธ์
002	ฝ่ายบุคคล
003	ฝ่ายการตลาด

4. สร้างวิว Gemployee ประกอบด้วย EMP_ID, NAME, SURNAME และ DEPTNAME สำหรับ DEPT_ID เท่ากับ 001

การบ้าน 9

1. กำหนดให้ ทรานแซกชันที่ 2 มีการเรียกใช้ข้อมูลที่ถูกปรับแก้โดยทรานแซกชันที่ 1 ซึ่งมีปัญหา และจะต้องถูกยกเลิกเพื่อเริ่มต้นทำงานนั้นใหม่ ในการนับจำนวนวิชาที่นักศึกษาคนหนึ่งลงทะเบียน ดังนี้

ทรานแซกชันที่ 1	เวลา	ทรานแซกชันที่ 2	ค่าของข้อมูลในฐานข้อมูล
อ่านจำนวนวิชา	Time 1		6
เพิ่ม 2 รายวิชา	Time 2		6
บันทึกจำนวนวิชา	Time 3		8
	Time 4	อ่านจำนวนวิชา	8
	Time 5	ถอน 3 รายวิชา	8
มีปัญหาและต้องเริ่มต้นใหม่	Time 6		6
	Time 7	บันทึกจำนวนวิชา	5

แสดงการแก้ไขการทำงานข้างต้นด้วยวิธีการจัดลำดับความขัดแย้ง (Conflict Serializability)

2. กำหนดคำสั่งในการเรียกดู เพิ่มและแก้ไขข้อมูลจำนวนเงินในบัญชีรายการ 1 และรายการ 2 ดังนี้

รายการ 1	รายการ 2
SELECT SUM (จำนวนเงิน) FROM บัญชีออมทรัพย์;	UPDATE บัญชีออมทรัพย์ SET จำนวนเงิน = จำนวนเงิน + 2000 WHERE เลขที่บัญชี = 'A0002';
	INSERT INTO บัญชีออมทรัพย์ VALUES ('A0005', 3000)

กำหนดฐานข้อมูล "บัญชีออมทรัพย์" ดังนี้

เลขที่บัญชี	จำนวนเงิน
A001	10000
A002	2500
A003	3000
A004	5000

แสดงวิธีการจัดลำดับความขัดแย้ง (Conflict Serializability)

3. เขียนคำสั่ง SQL ในการกำหนดสิทธิการใช้งานของผู้ใช้ในแต่ละข้อต่อไปนี้

3.1 กำหนดสิทธิในการเรียกดู แก้ไข และเพิ่มข้อมูลให้ Duangjai สำหรับตาราง STUDENTS

3.2 กำหนดให้ทั้ง Duangjaii และ Nipa สามารถแก้ไขและลบข้อมูล ในตาราง COURSES ได้

3.3 กำหนดให้ Nipa มีสิทธิเปลี่ยนค่าในคอลัมน์ COURSENO ในตาราง COURSES

3.4 กำหนดให้ Nipa สามารถทำคำสั่งใด ๆ ในตาราง COURSES ได้