

Data concurrency and locking

01418221 Fundamentals of Database Systems

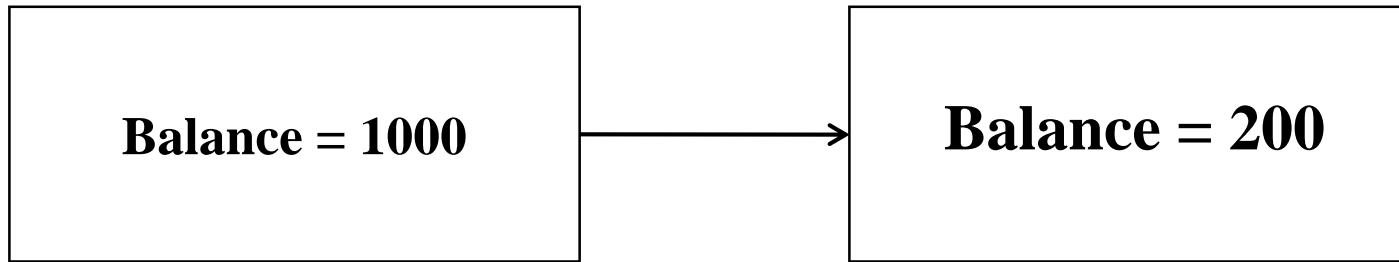
Outlines

- Transactions
- Concurrency & Locking
- Lock Wait
- Deadlocks

What is a transaction?

Your bank account

Your Mom's bank account



- โอนเงิน 100 บาทจากบัญชีของคุณไปยังบัญชีแม่:
- หักเงิน 100 บาทจากบัญชีธนาคารของคุณ (ลบ 100 บาท)
- เพิ่ม 100 บาทเข้าบัญชีธนาคารของแม่ (เพิ่ม 100 บาท)

What is a transaction?

- เป็นการรวมคำสั่ง SQL มากกว่าหนึ่งคำสั่งเป็นหนึ่งการทำงานเดียวกัน (a single unit หรือ Unit of Work (UOW)
- ธุรกรรม (Transaction) เริ่มต้นด้วยคำสั่ง SQL ไต ๆ และลงท้ายด้วย **COMMIT** หรือ **ROLLBACK**
 - คำสั่ง COMMIT ทำการเปลี่ยนแปลงอย่างถาวรไปยังฐานข้อมูล
 - คำสั่ง ROLLBACK ยกเลิกการทำงานโดยการย้อนกลับการเปลี่ยนแปลงไปอยู่ค่าเดิม
 - คำสั่ง COMMIT และ ROLLBACK จะปลดล็อคข้อมูลทั้งหมด

Example of transactions

First SQL statement
starts transaction

INSERT INTO employee VALUES (100, 'JOHN')
INSERT INTO employee VALUES (200, 'MANDY')
COMMIT

empID	Name
100	JOHN
200	MANDY

DELETE FROM employee WHERE name = 'MANDY'
UPDATE employee SET empID = 101 where name = 'JOHN'
ROLLBACK

No changes applied due
to ROLLBACK

empID	Name
100	JOHN
200	MANDY

UPDATE employee SET name = 'JACK' where empID= 100
COMMIT

empID	Name
100	JOHN
200	MANDY

ROLLBACK **X**

Transactions – ACID rules

- **Atomicity (ครบหน่วย)**

- คำสั่งทั้งหมดในทรานแซกชันจะปฏิบัติเสมือนชุดคำสั่งเดียวกัน
- ถ้าทรานแซกชันทำงานเสร็จสมบูรณ์ การเปลี่ยนแปลงข้อมูลจะถูกบันทึกลงในฐานข้อมูล (committed)
- ถ้าทรานแซกชันทำงานล้มเหลว การเปลี่ยนแปลงดำเนินการก่อนจุดคำสั่งที่ล้มเหลวจะถูกยกเลิกทั้งหมด (rolled back)

- **Consistency (ความสอดคล้องตรงกัน)**

- ทรานแซกชันจะนำข้อมูลจากสถานะที่สอดคล้องกันไปยังอีกสถานะหนึ่งดังนั้นข้อมูลที่สอดคล้องที่ถูกต้องเท่านั้นจึงจะถูกจัดเก็บในฐานข้อมูล

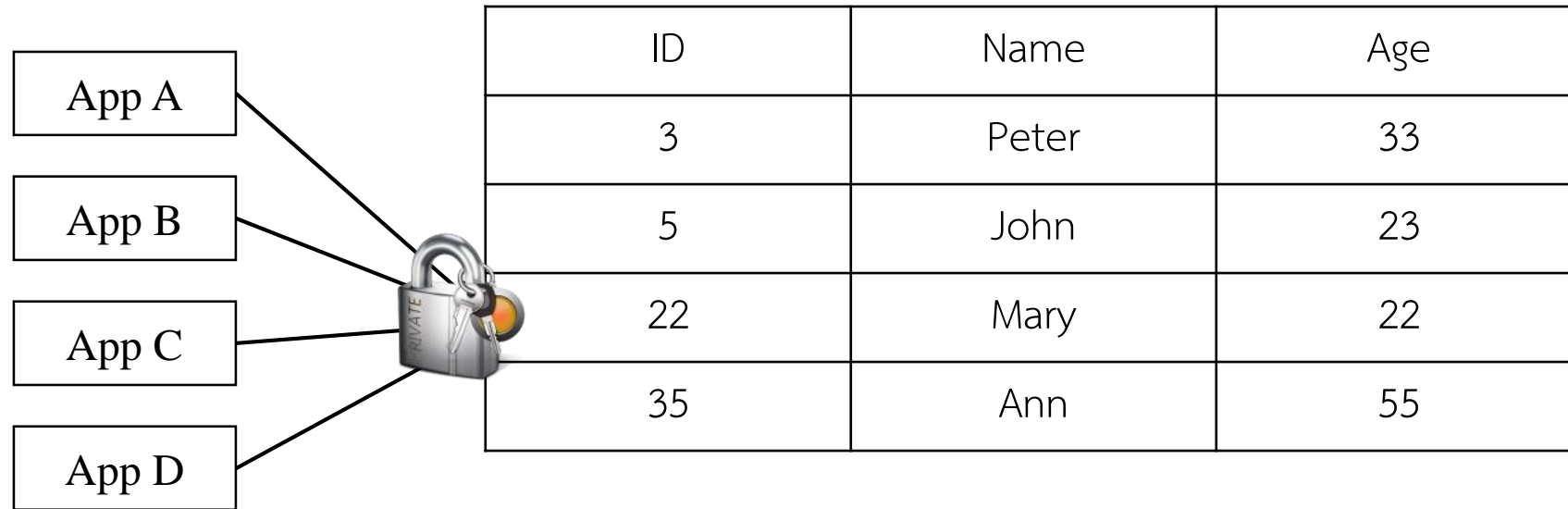
- **Isolation (ไม่ถูกแทรกแซง)**

- ธุรกรรมที่เกิดขึ้นพร้อมกันไม่สามารถแทรกแซงซึ่งกันและกัน

- **Durability (การบันทึกอย่างถาวร)**

- ทรานแซกชันที่ยืนยันการเปลี่ยนแปลงแล้วจะต้องเปลี่ยนแปลงข้อมูลในฐานข้อมูลจริง

Concurrency and Locking



- **Concurrency:**
 - ผู้ใช้หลายคนเข้าถึงทรัพยากรเดียวกันในเวลาเดียวกัน
- **Locking:**
 - กลไกเพื่อรับรองความถูกต้องและความสอดคล้องของข้อมูล

Locking

- ล็อกจะได้รับโดยอัตโนมัติตามความจำเป็นเพื่อรองรับการทำ transaction ตามลักษณะ “isolation levels”
- คำสั่ง COMMIT and ROLLBACK จะปลดล็อกข้อมูลทั้งหมด
- การล็อกแบ่งออกเป็น 2 ประเภทคือ
 1. Share Lock (S locks) ได้มาเมื่อแอปพลิเคชันต้องการอ่านและป้องกันไม่ให้อื่นอัปเดตแถวเดียวกัน
 2. Exclusive locks (X locks) ได้มาเมื่อแอปพลิเคชันต้องการเพิ่ม แก้ไข และลบข้อมูล

Problems if there is no concurrency control

- Lost update
- Uncommitted read
- Non-repeatable read
- Phantom read

Lost update

reservations

Seat	Name	...
7C	Mary	
7B		
...		

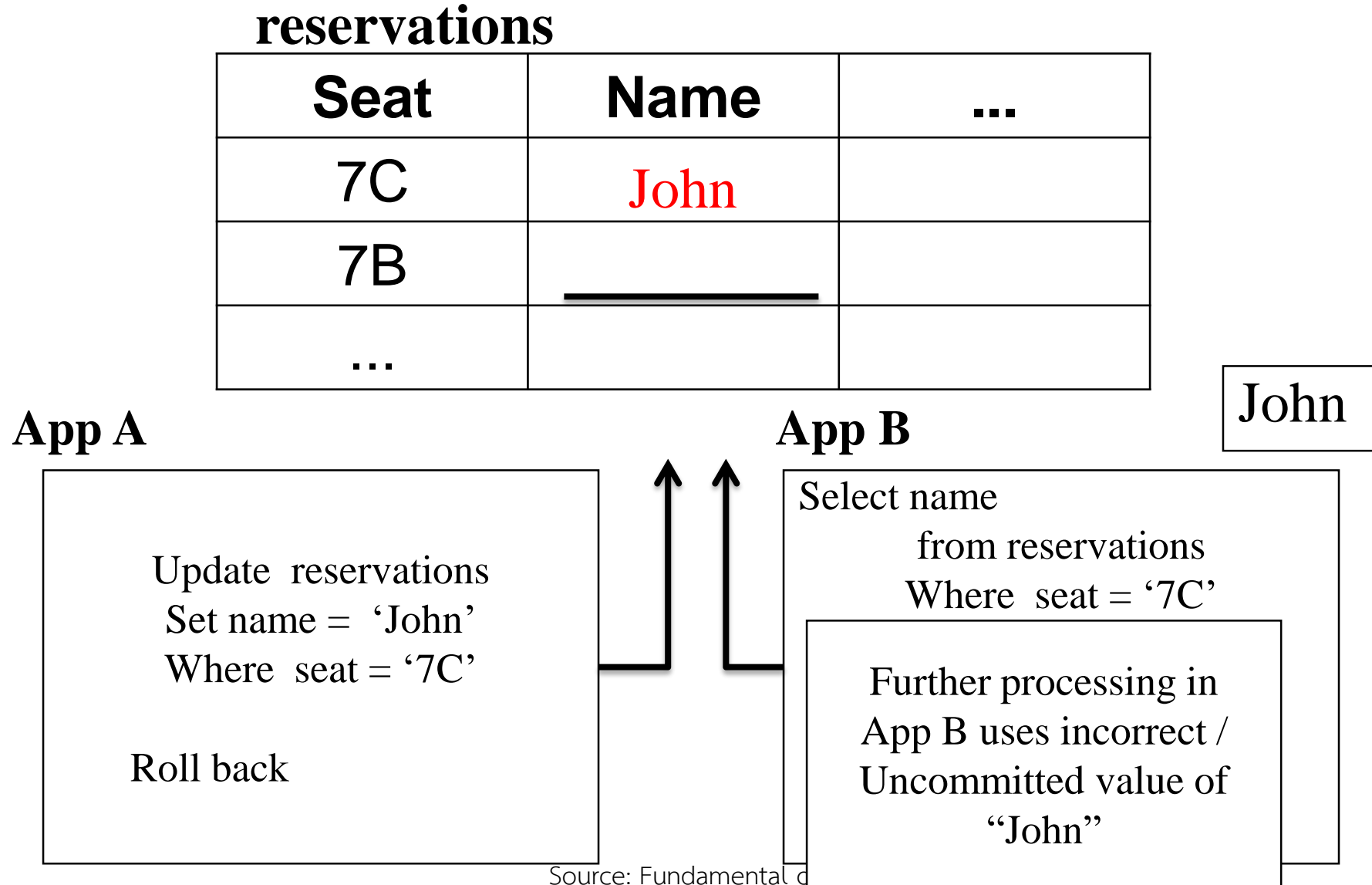
App A

Update reservations
Set name = 'John'
Where seat = '7C'

App B

Update reservations
Set name = 'Mary'
Where seat = '7C'

Uncommitted read (also known as “dirty read”)



Non-repeatable read

reservations

Seat	Name	...
7C	John	
7B		
...		

App A

7C

7B

7B

Select seat
From reservations
Where name is
NULL

...
Select seat
From reservations
Where name is
NULL

App B

Update reservations
set name = 'John'
Where seat = '7C'

The same SELECT (read) returns a
Different result : Less rows (in this
case '7C' doesn't show anymore).
This is a non- repeatable read

amental of

Phantom read

Seat	Name	...
7C	Susan	
7B		
...		

App A

7C

7C
7B

Select seat
From reservations
Where name is
NULL

...
Select seat
From reservations
Where name is
NULL

App B

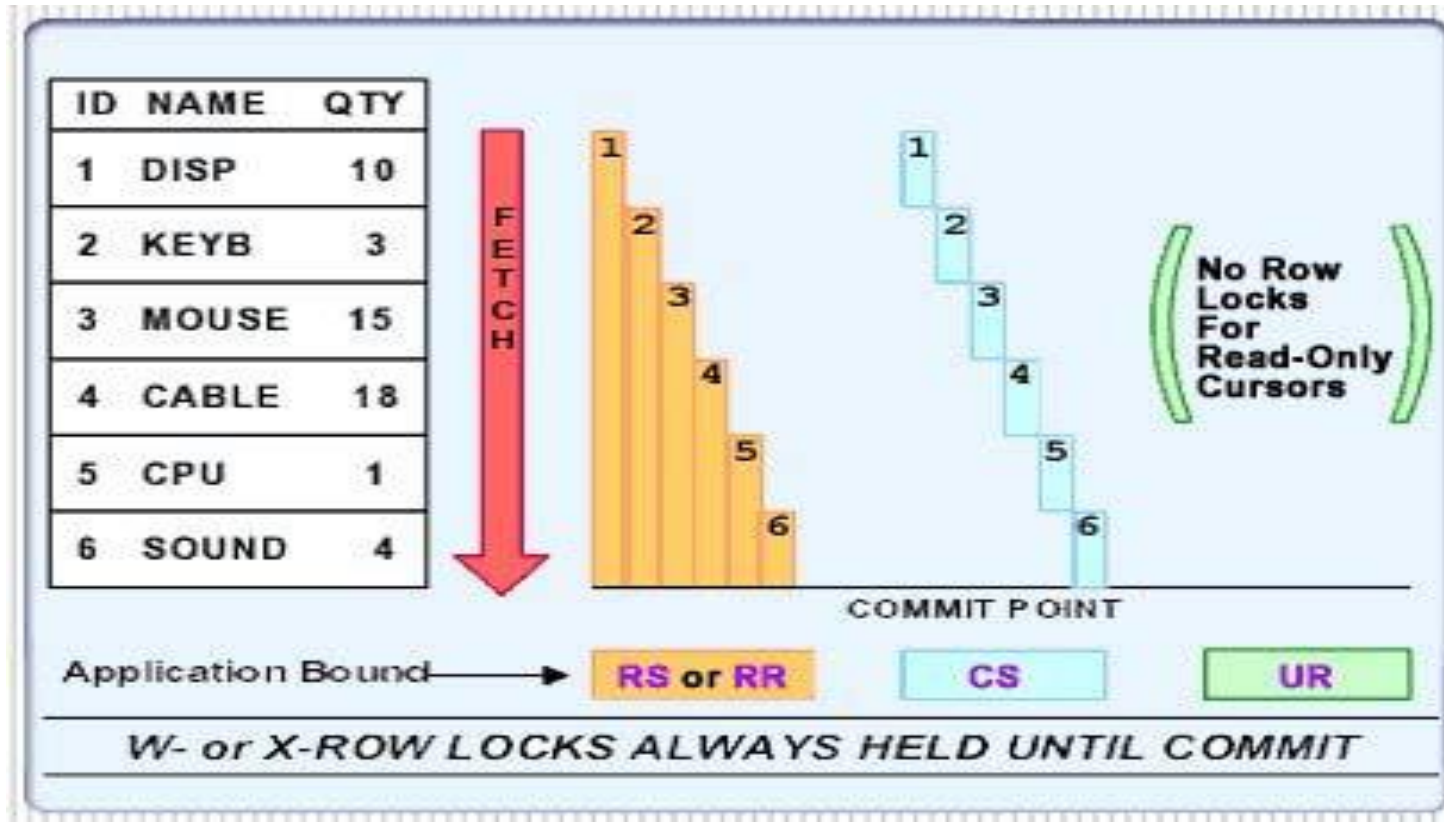
Update reservations
set name = NULL
Where seat = '7C'

The same SELECT (read) returns a
Different result : More rows (phantom rows,
in this case '7C' doesn't show)
This is a phantom read

Isolation levels

- “นโยบาย (Policies)” ในการควบคุมการล็อกข้อมูล
- **DB2** ให้ระดับการควบคุมที่แตกต่างกันเพื่อให้มีการบันทึกข้อมูลไม่กระทบต่อกัน
 - Uncommitted Read (UR)
 - Cursor Stability (CS)
 - Currently committed (CC)
 - Read Stability (RS)
 - Repeatable Read (RR)

Comparing isolation levels

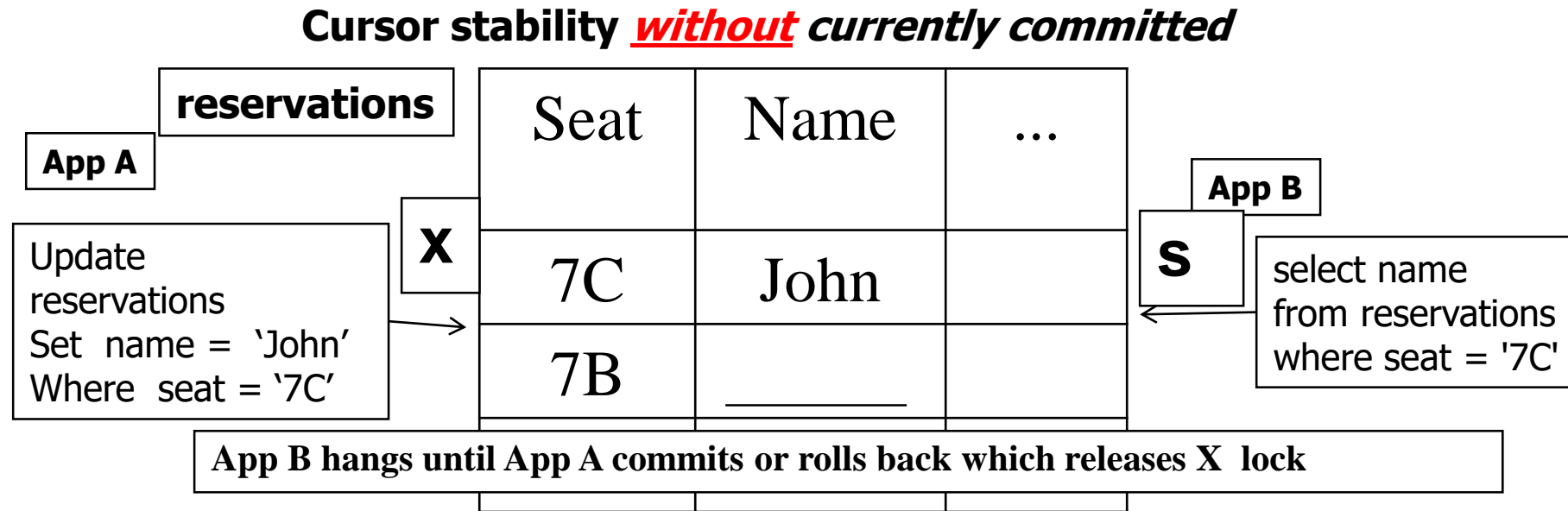


Cursor stability with currently committed (CC) semantics

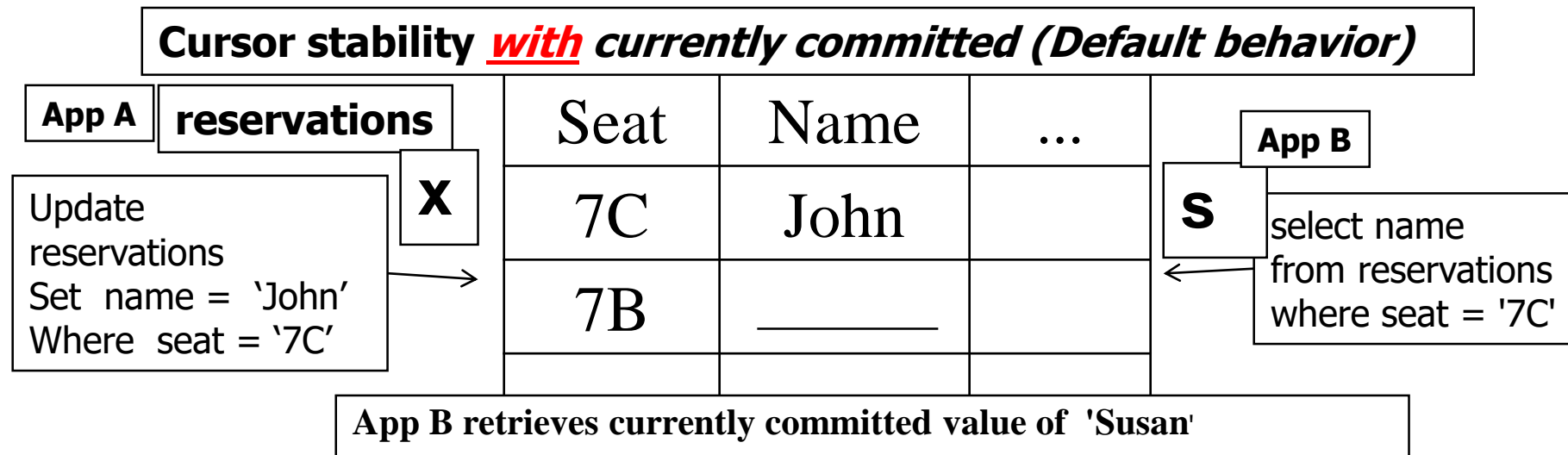
- Avoids timeouts and deadlocks

Cursor stability		Cursor stability with currently committed	
Situation	Result	Situation	Result
Reader blocks reader	NO	Reader blocks reader	NO
Reader blocks Writer	Maybe	Reader blocks Writer	No
Writer block Reader	Yes	Writer block Reader	No
Writer blocks Writer	Yes	Writer blocks Writer	Yes

Cursor stability without currently committed (CC) semantics



Cursor stability with currently committed (CC) semantics



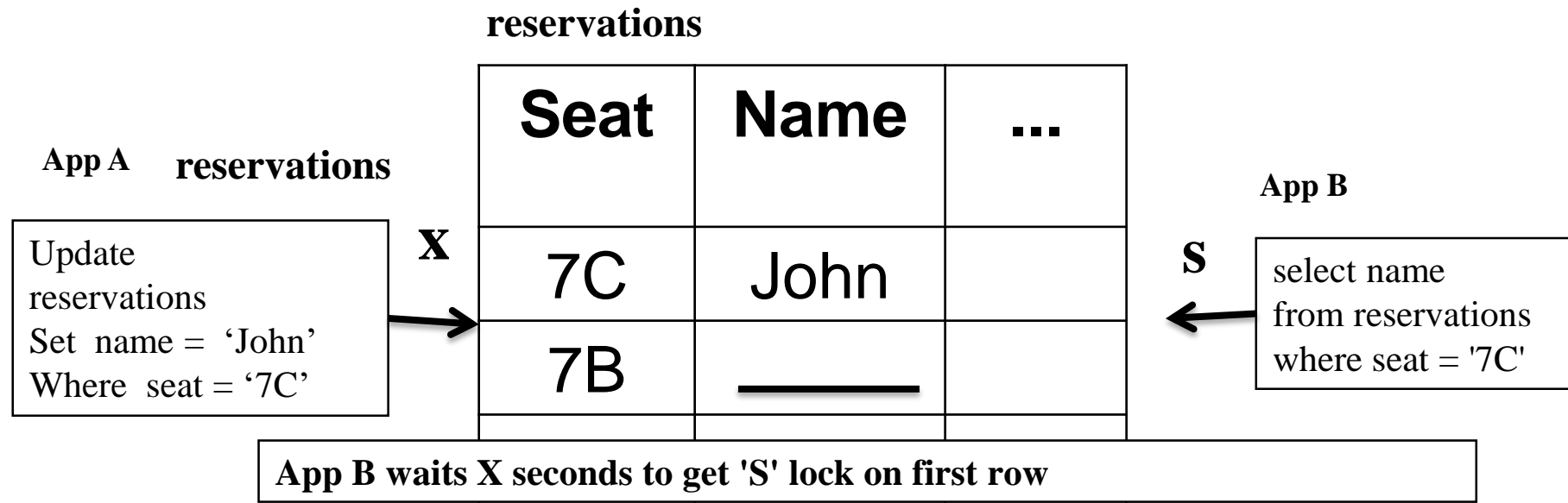
Comparing and choosing an isolation level

Isolation Level	Lost update	Dirty Read	Non-repeatable Read	Phantom Read
Repeatable Read (RR)	-	-	-	-
Read Stability (RS)	-	-	-	Possible
Cursor Stability (CS)	-	-	Possible	Possible
Uncommitted Read (UR)	-	Possible	Possible	Possible

Application Type	High data stability required	High data stability not required
Read-write RS CS transactions	RS	CS
Read-only RS or RR UR transactions	RS or RR	UR

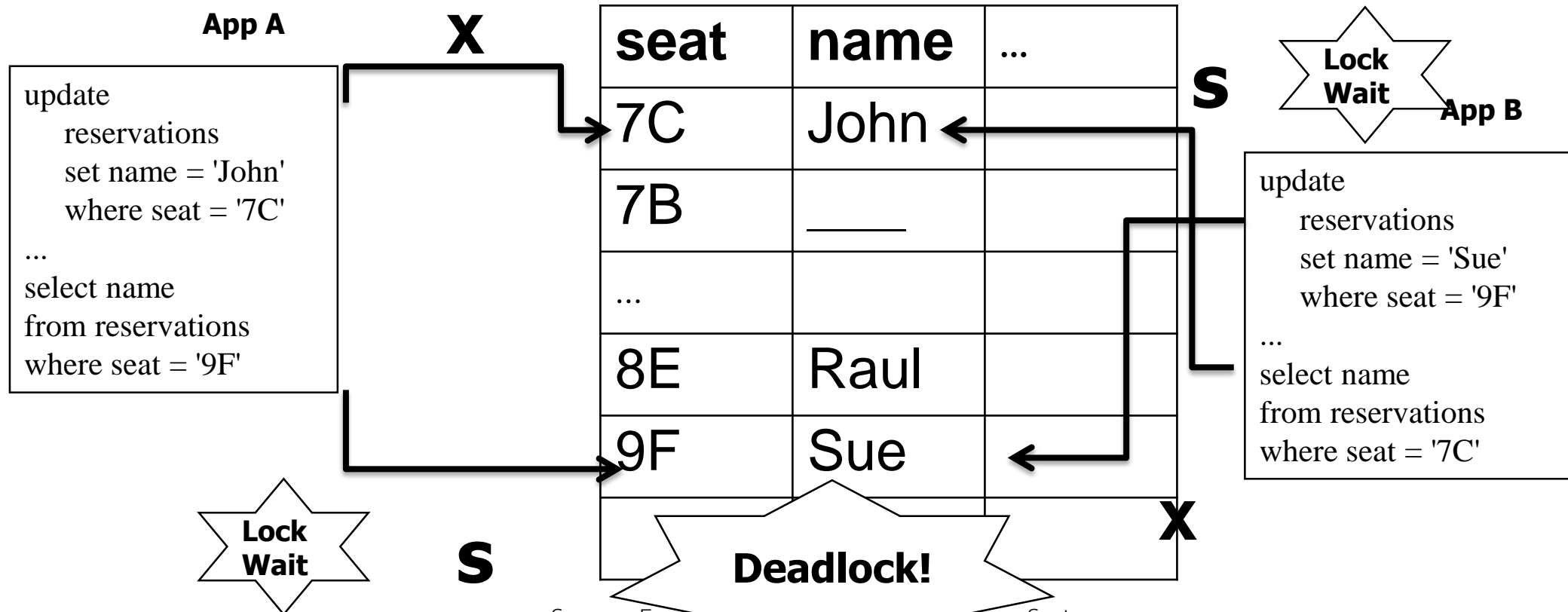
Lock wait

- โดยทั่วไปแอปพลิเคชันจะรออย่างไม่มีกำหนดเพื่อรับการล็อคที่จำเป็นใช้ในการทำงาน



Deadlocks

- จะเกิดขึ้นเมื่อแอปพลิเคชันสองตัวรอคอยทรัพยากรเดียวกัน
- แต่ละแอปพลิเคชันจะถือครองทรัพยากรที่อีกฝ่ายหนึ่งต้องการอยู่
- การรอคอยจะไม่สามารถแก้ไขได้ด้วยตัวเอง (การลือคตาย)



Deadlocks

- การลือคตายมักจะเกิดขึ้นจากการออกแบบแอปพลิเคชันที่ไม่ดีพอ
- เมื่อเกิด deadlock ขึ้นแล้ว DBMS จะเรียกโปรแกรมจัดการนี้ คือ
 - เลือก transaction ที่ต้องการทำ roll back ในขณะที่อีกตัวหนึ่งจะได้ทำงานต่อไป
 - transaction ที่ถูกให้ยกเลิกการทำงานจะแสดงผลลัพธ์ออกมาในลักษณะ SQL error
 - กระบวนการ rollback จะทำให้มีการปลดล็อกในทุกส่วน