CANTHO UNIVERSITY

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN

QUẢN TRỊ DỮ LIỆU - CT467

Chương 3: GIAO DỊCH (Transaction)

Biên soạn:

Ths. Nguyễn Thị Kim Yến

Ntkyen@ctu.edu.vn

CANTHO UNIVERSITY

MỤC TIÊU CỦA CHƯƠNG 3

- Giới thiệu các nguyên tắc xử lý giao dịch trong hệ quản trị CSDL bao gồm:
  - Các khái niệm có liên quan đến xử lý giao dịch
  - Điều khiển sự cạnh tranh giữa các giao dịch
  - Và một số tính chất của các lịch trình

CANTHO UNIVERSITY

NỘI DUNG

1. Khái niệm giao dịch
2. Các trạng thái của giao dịch
3. Cạnh tranh giao dịch
4. Lịch trình giao dịch
5. Tính khả tuần tự
6. Tính phục hồi

1. Khái niệm

- **Giao dịch** (GD): một tập các chỉ thị trong hệ QT CSDL nhằm thực hiện một tác vụ hay một chức năng nào đó.

Góc độ người dùng

- Tác vụ
- Chức năng duy nhất

Góc độ hệ QT CSDL

- Nhiều chỉ thị, nhiều thao tác
- Một số chỉ thị cập nhật CSDL

1. Khái niệm (tt)

❖ **Ví dụ:** Chuyển khoản một số tiền X đồng từ tài khoản A sang tài khoản B là một giao dịch

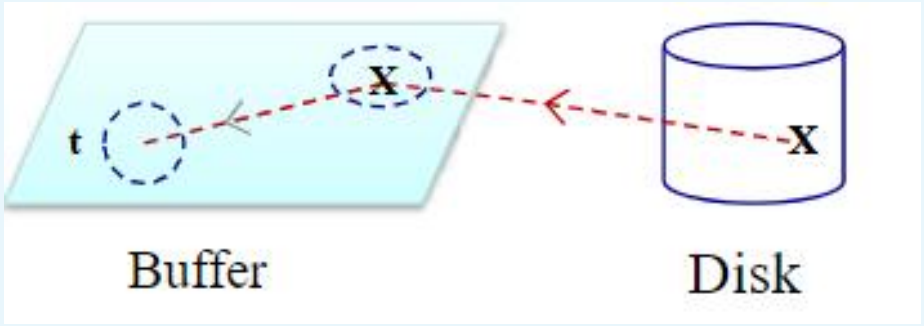
❖ **Giao dịch T :**

- Read (A)
- A = A – X
- Write (A)
- Read (B)
- B = B + X
- Write (B)

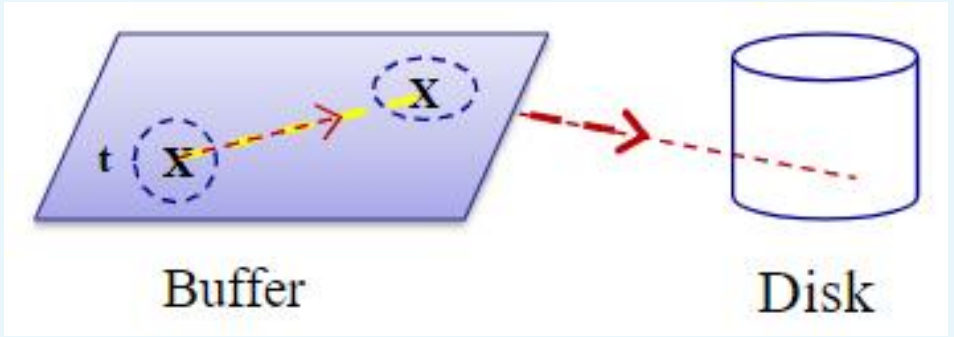
Các thao tác của giao dịch

❖Các truy xuất trong CSDL được thực hiện bởi 2 hoạt động:

- READ(X): đọc 1 hạng mục X từ CSDL vào vùng nhớ GD
- WRITE(X): ghi hạng mục X từ vùng nhớ GD vào CSDL



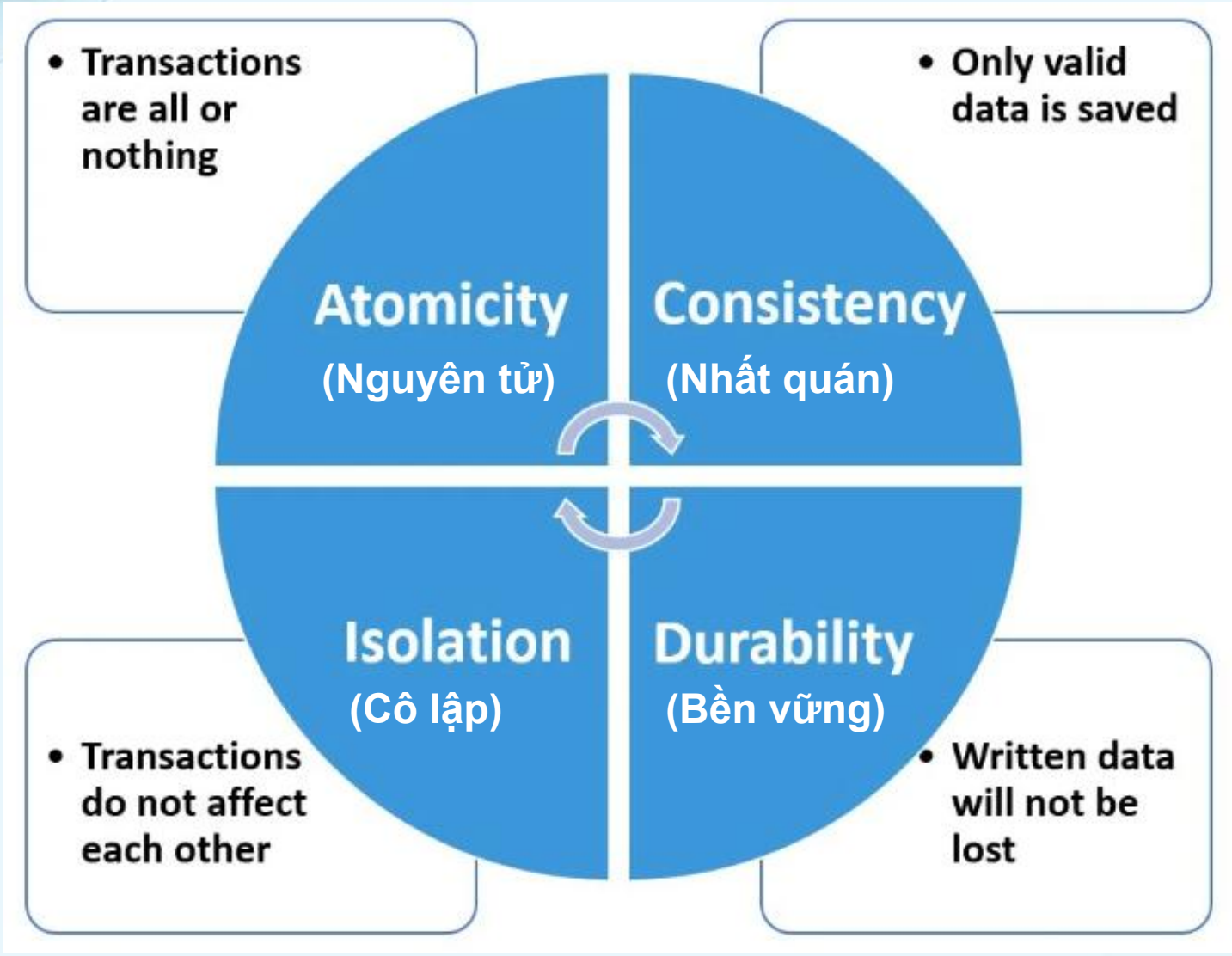
Read (X, t)



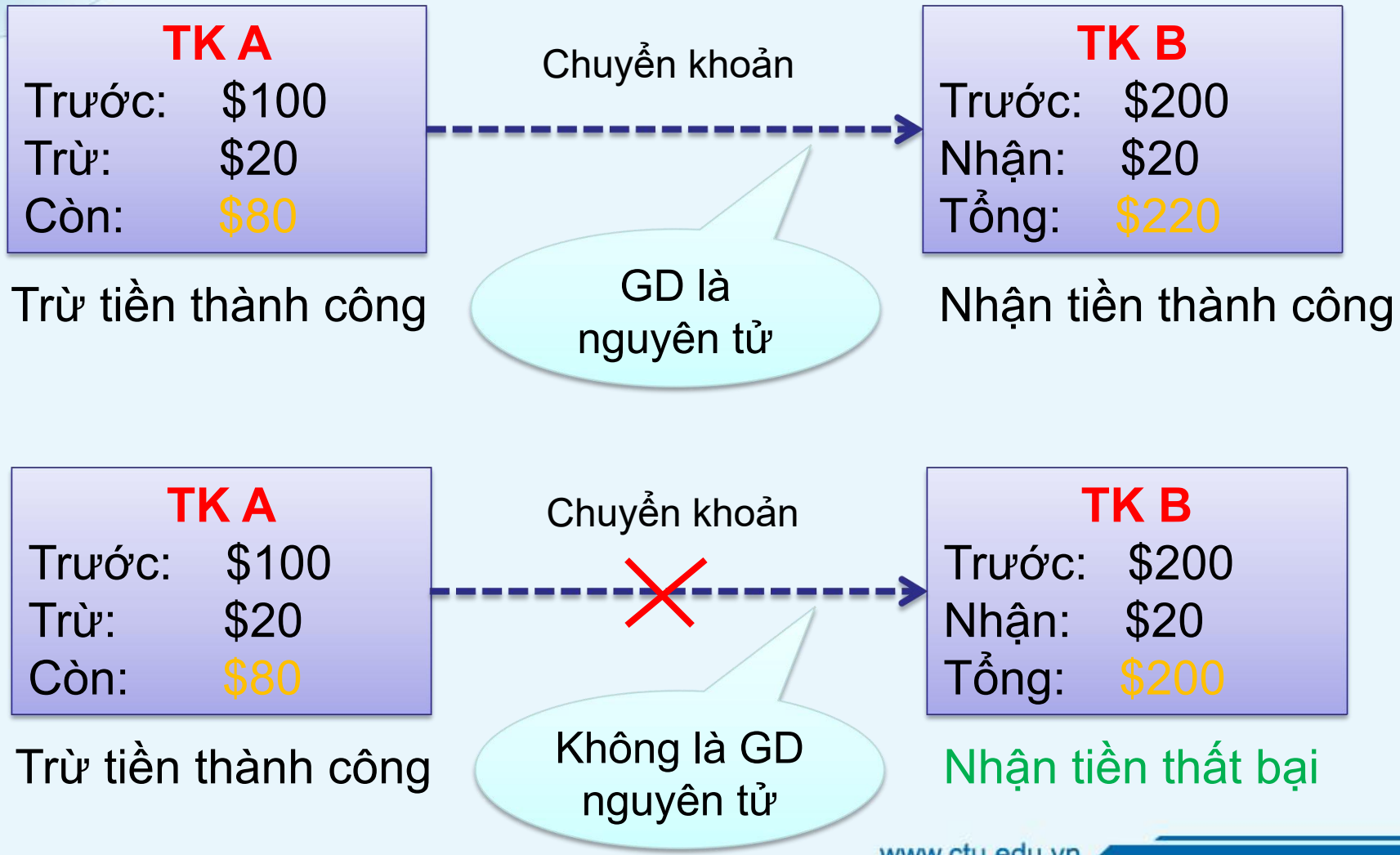
Write (X, t)

CANTHO UNIVERSITY

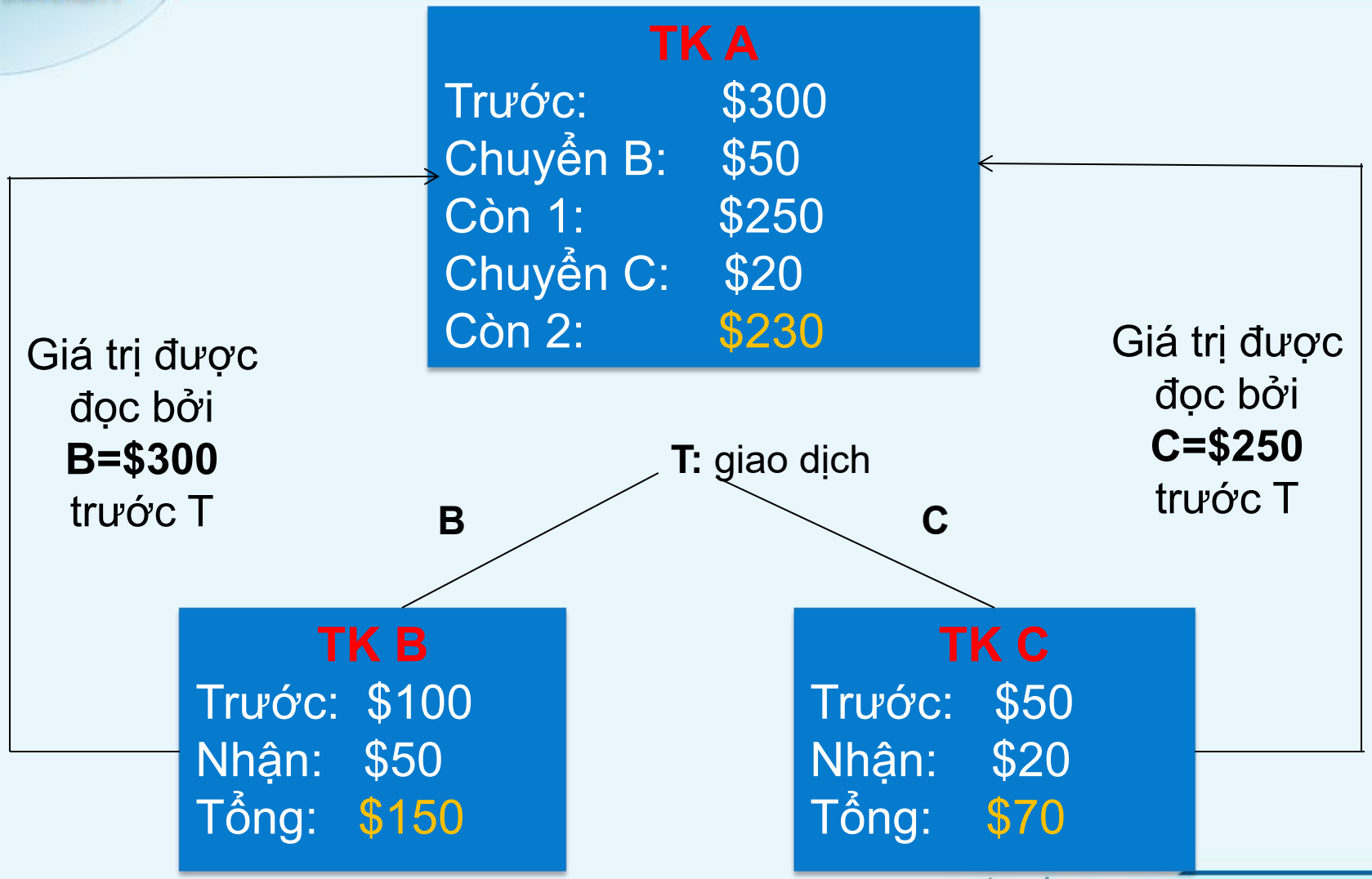
4 tính chất của giao dịch



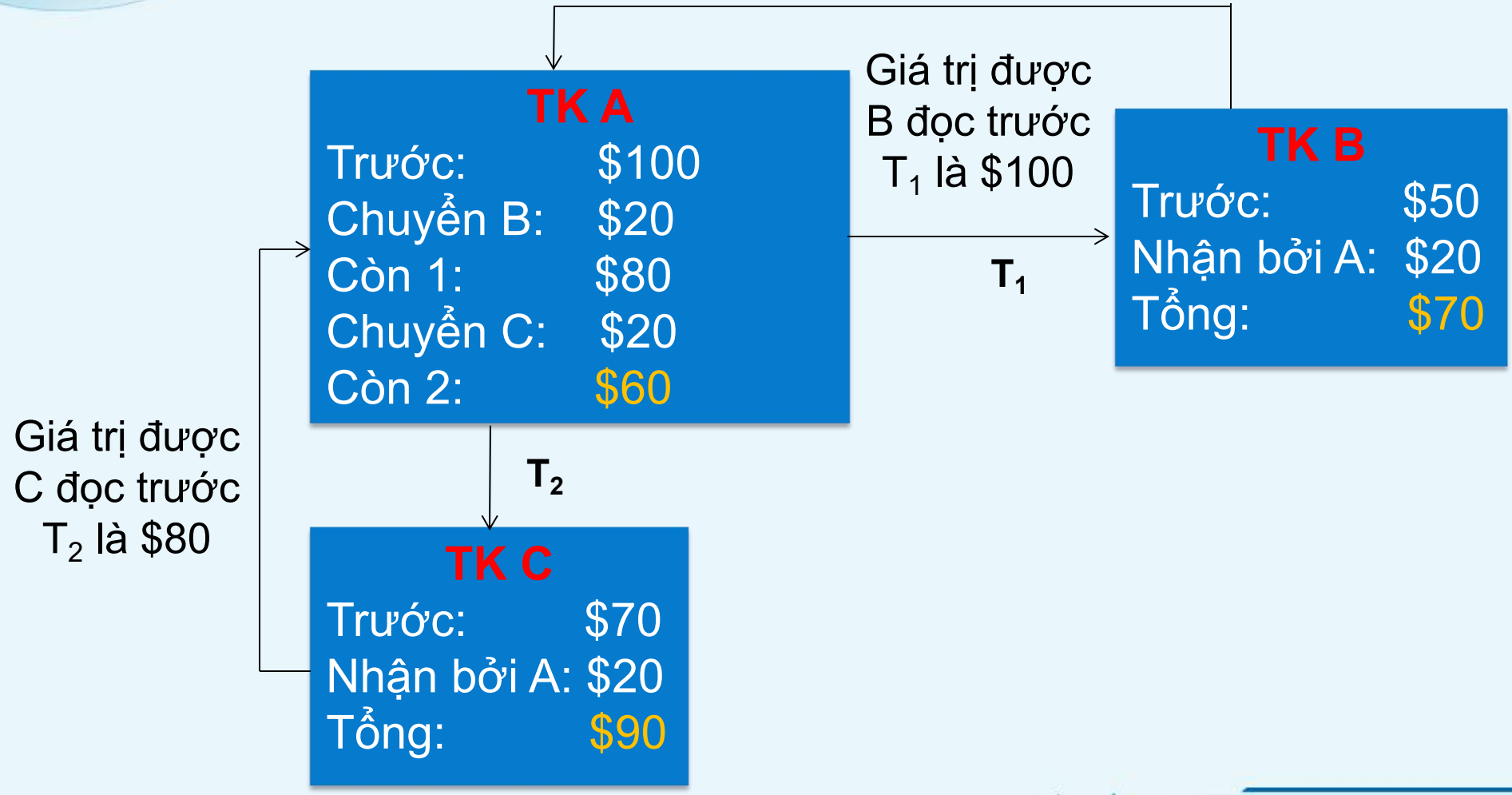
❖ Tính nguyên tử



❖ Tính nhất quán



❖ Tính cô lập


TK A
Trước: $100
Chuyển B: $20
Còn 1: $80
Chuyển C: $20
Còn 2: $60
Giá trị được B đọc trước T_1 là $100
T_1
TK B
Trước: $50
Nhận bởi A: $20
Tổng: $70
T_2
TK C
Trước: $70
Nhận bởi A: $20
Tổng: $90
Giá trị được C đọc trước T_2 là $80

❖ Tính bền vững

Chuyển khoản

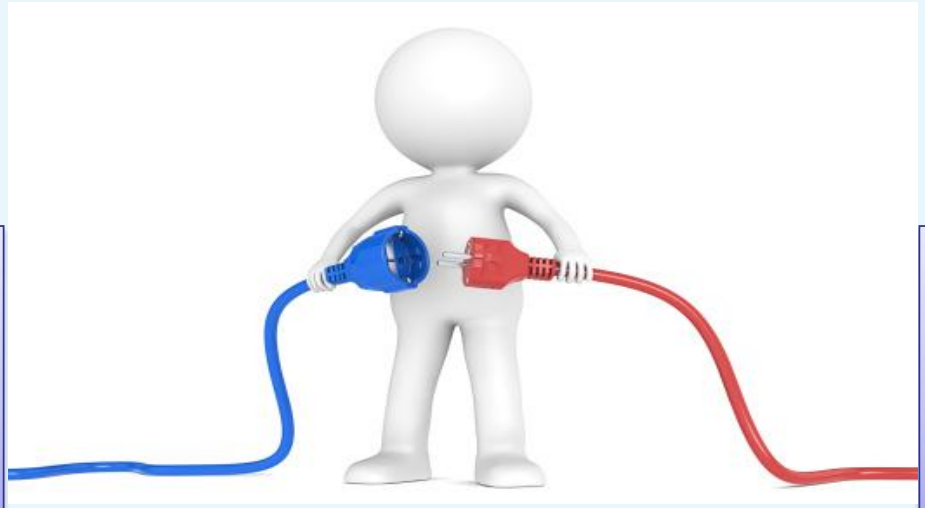
Thành công

TK A

Trước: $100

Trừ: $20

Còn: $80



TK B

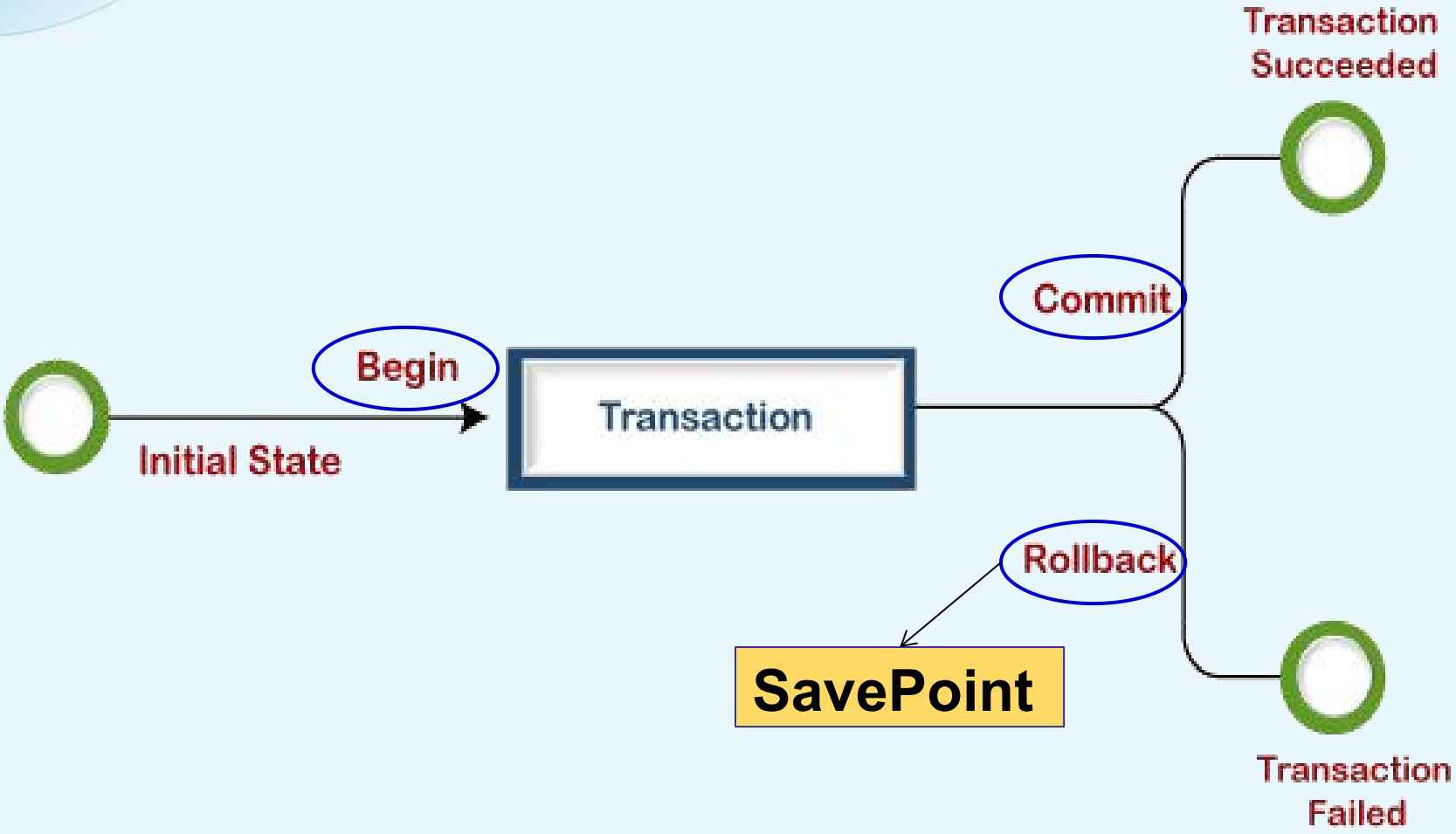
Trước: $200

Nhận: $20

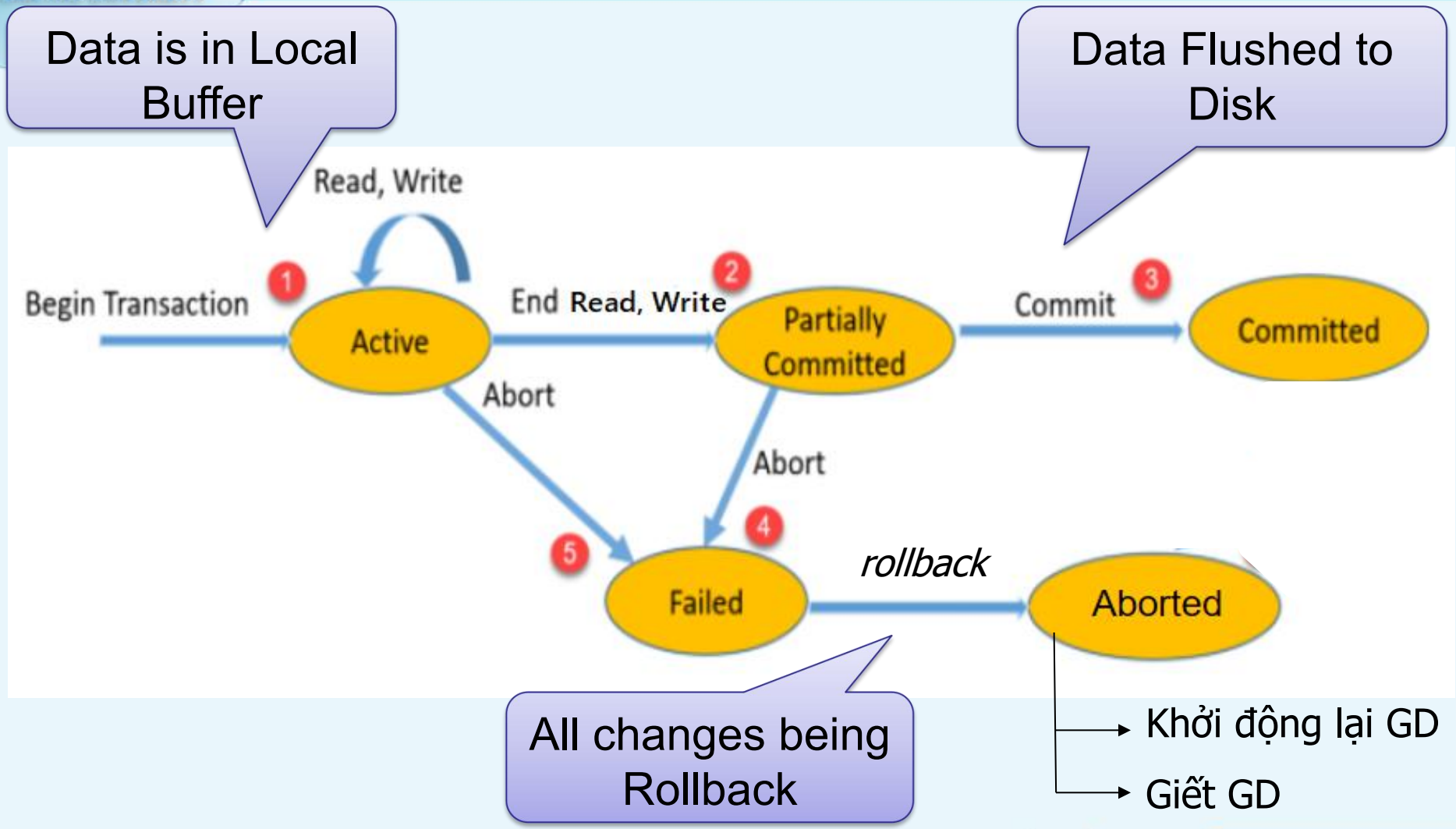
Tổng: $220

Hệ thống xãy ra sự cố

Cấu trúc của Transaction



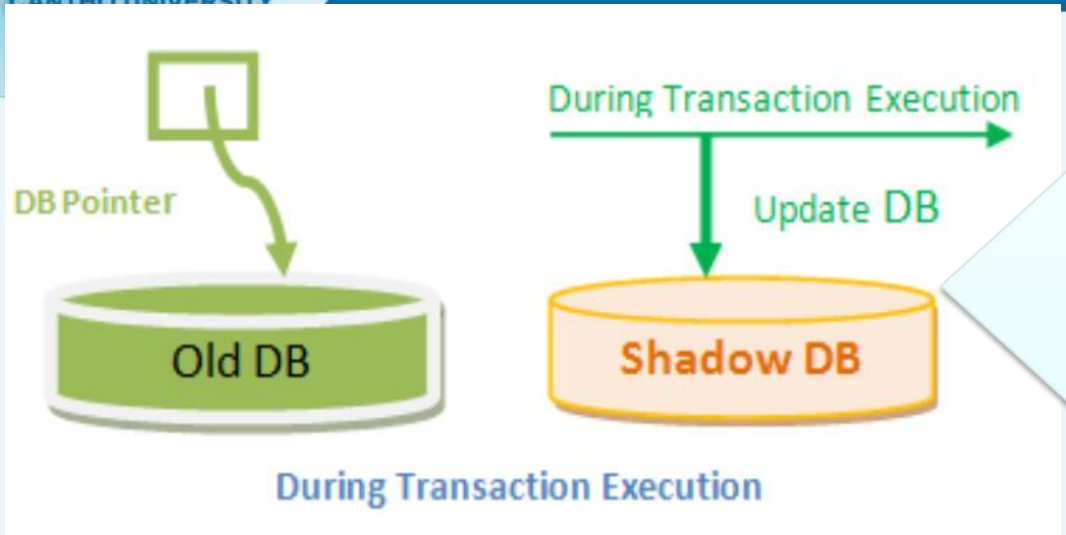
2. Các trạng thái của giao dịch (tt)



2. Các trạng thái của giao dịch (tt)

- **Hoạt động (Active):** GD sẽ duy trì trạng thái này trong khi đang thực hiện
- **Cam kết một phần (Partially Committed)**: sau khi thao tác cuối cùng trong GD được thực hiện
- **Thất bại (Failed)**: Sau khi phát hiện rằng sự thực hiện không thể tiếp tục được nữa
- **Bỏ dở (Aborted)**: Sau khi rollback và CSDL đã phục hồi lại trạng thái của nó trước khi khởi động GD
- **Cam kết (Committed)**: GD hoàn tất thành công

Thực thi tính nguyên tử và bền vững

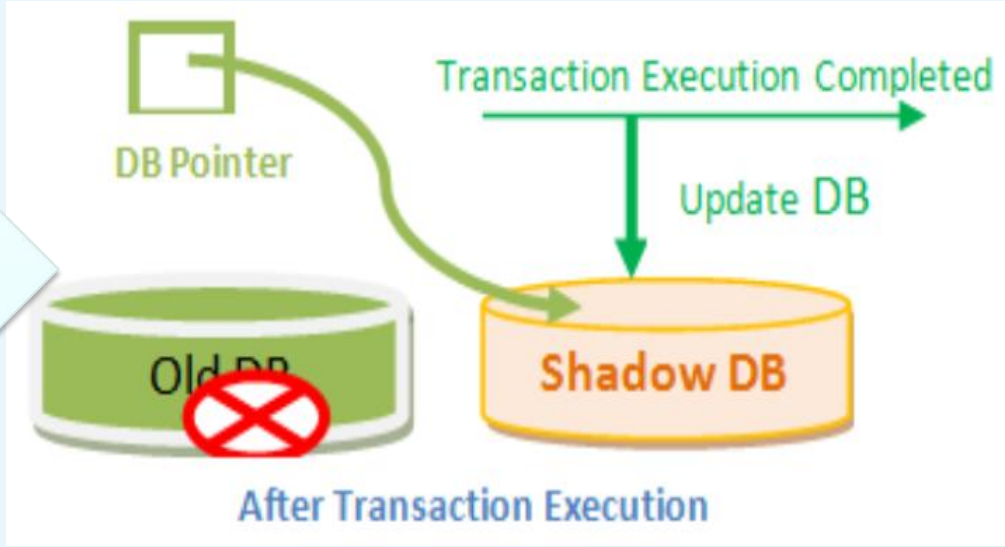


Bản sao bóng

(shadow copy)

Tất cả các thay đổi của các GD được cập nhật trong bản sao bóng của CSDL

Khi tất cả giao dịch hoàn tất, con trỏ DB thực hiện trỏ đến CSDL bóng ➔ bản sao mới của DB. Bản sao cũ của DB sẽ bị xóa



Thực thi tính nguyên tử và bền vững (tt)

❖ **Yêu cầu:**

- Sự cập nhật DB_pointer là nguyên tử (hoặc tất cả các byte của nó được viết hoặc không byte nào được viết)
- Đảm bảo bởi việc thực thi bản sao bóng của thành phần quản trị phục hồi (sao lưu toàn bộ CSDL)
- Không cho phép GD thực hiện đồng thời với các GD khác

➔ Cực kỳ thiếu hiệu quả trong ngữ cảnh **CSDL lớn**

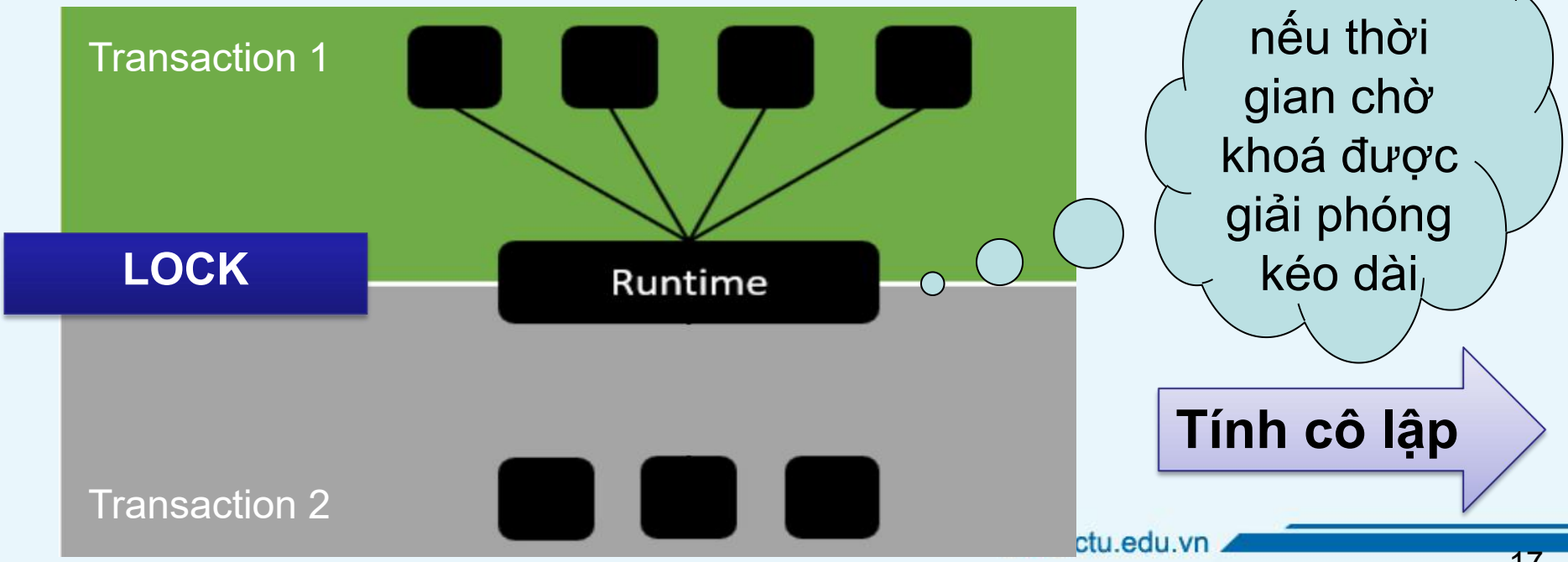
3. Cạnh tranh giao dịch

- Khi 2 GD thực hiện đồng thời khó khăn trong việc đảm bảo **tính nhất quán** của dữ liệu.

GD1	GD2	Nhận xét
Read	Read	Không có tranh chấp
Read	Write	Xảy ra tranh chấp
Write	Read	Xảy ra tranh chấp
Write	Write	Chỉ cho phép có đúng 1 GD được ghi trên đơn vị dữ liệu tại một thời điểm

3. Cạnh tranh giao dịch (tt)

- Để giải quyết vấn đề nêu trên, hệ QTCSDL sử dụng cơ chế khóa (locking) ➔ quyết định GD nào được thực hiện trước và GD nào phải chờ.

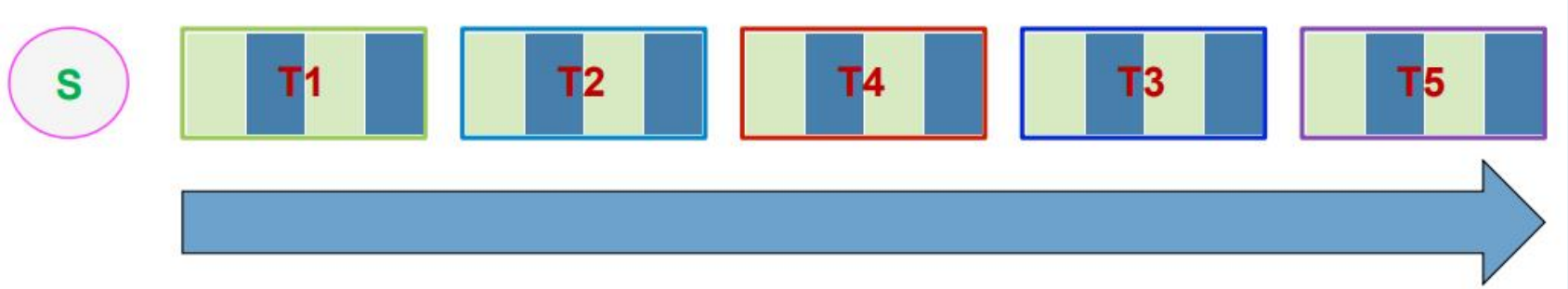


4. Lịch trình (Schedule) giao dịch

- **Lịch trình GD:** một dãy (có thứ tự) các thao tác của một tập các GD mà trong đó thứ tự của các thao tác trong mỗi GD được bảo toàn.
  - Lịch trình tuần tự
  - Lịch trình cạnh tranh

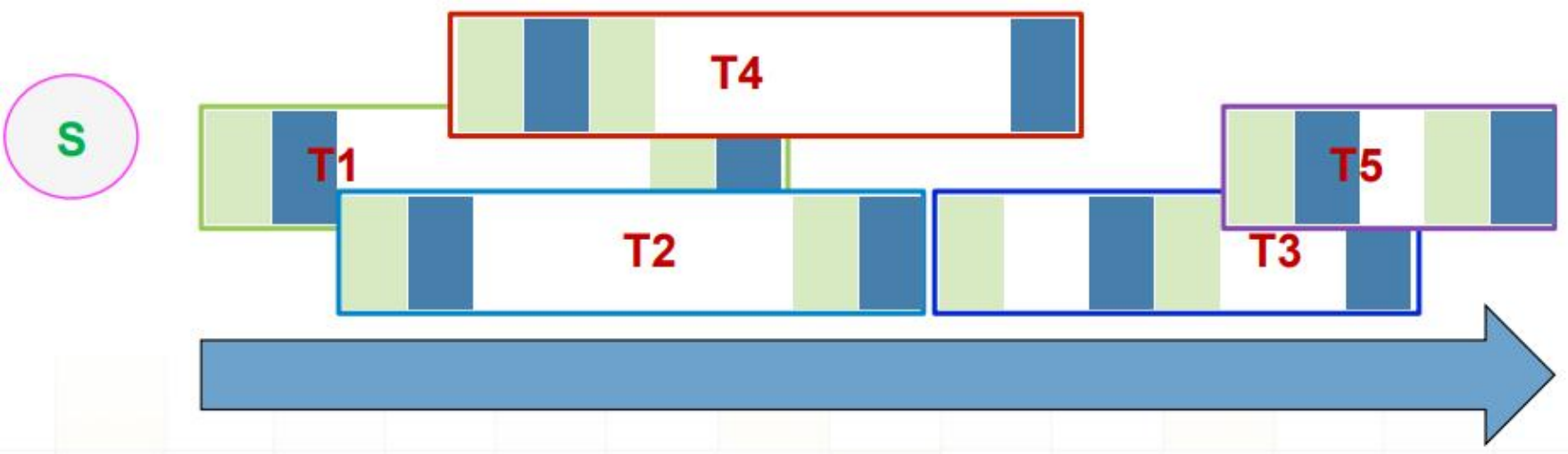
4. Lịch trình (Schedule) giao dịch (tt)

– **Lịch trình tuần tự:** các thao tác được thực hiện liên tiếp tiếp nhau, không có thao tác GD khác xen vào (n!)



4. Lịch trình (Schedule) giao dịch (tt)

- **Lịch trình cạnh tranh:** là lịch trình trong đó các GD thực hiện đan xen nhau/đồng thời với nhau (> n!)



4. Lịch trình giao dịch (tt)

- **Ví dụ:** Giả sử T1 và T2 là hai GD chuyển khoản
  - T1: chuyển 50$ từ tài khoản A sang tài khoản B
  - T2: chuyển 10% số dư từ TK A sang TK B

T1
R(A)
A = A - 50
W(A)
R(B)
B = B + 50
W(B)

T2
R(A)
temp = A*0.1
A = A - temp
W(A)
R(B)
B = B + temp
W(B)

- Giả sử giá trị hiện tại của:

A là 1000$ và B là 2000$

- Tổng A, B = **3000$**

❖ Lịch trình tuần tự

Tính nhất quán CSDL

T1	T2	A	B
R(A)			
A = A - 50			
W(A)		950	
R(B)			
B = B + 50			
W(B)			2050
	R(A)		
	temp = A*0.1		
	A = A - temp		
	W(A)	**855**	
	R(B)		
	B = B + temp		
	W(B)		**2145**

Schedule 1 3000

T1	T2	A	B
	R(A)		
	temp = A*0.1		
	A = A - temp		
	W(A)	900	
	R(B)		
	B = B + temp		
	W(B)		2100
R(A)			
A = A - 50			
W(A)		**850**	
R(B)			
B = B + 50			
W(B)			**2150**

Schedule 2 3000

❖ Lịch trình cạnh tranh

Trạng thái không nhất quán

T1	T2	A	B
R(A)			
A = A - 50			
W(A)		950	
	R(A)		
	temp = A*0.1		
	A = A - temp		
	W(A)	**855**	
R(B)			
B = B + 50			
W(B)			2050
	R(B)		
	B = B + temp		
	W(B)		**2145**

Schedule 3 3000

Lịch trình khả tuần tự (KQ tương đương 1 LT tuần tự)

T1	T2	A	B
R(A)			
A = A – 50			
	R(A)		
	temp = A*0.1		
	A = A - temp		
	W(A)	900	
	R(B)		
W(A)		**950**	
R(B)			
B = B + 50			
W(B)			2050
	B = B + temp		
	W(B)		**2100**

Schedule 4 3050

Lịch trình không khả tuần tự

5. Tính khả tuần tự (Serializability)

- Một lịch trình có tính khả tuần tự là một lịch trình tương đương với một lịch trình tuần tự nào đó
- Kết quả tương đương: phát sinh cùng trạng thái cuối của CSDL
- Tuy nhiên, KQ tương đương là chưa đủ để thể hiện sự tương đương của 2 lịch trình

5. Tính khả tuần tự (tt)

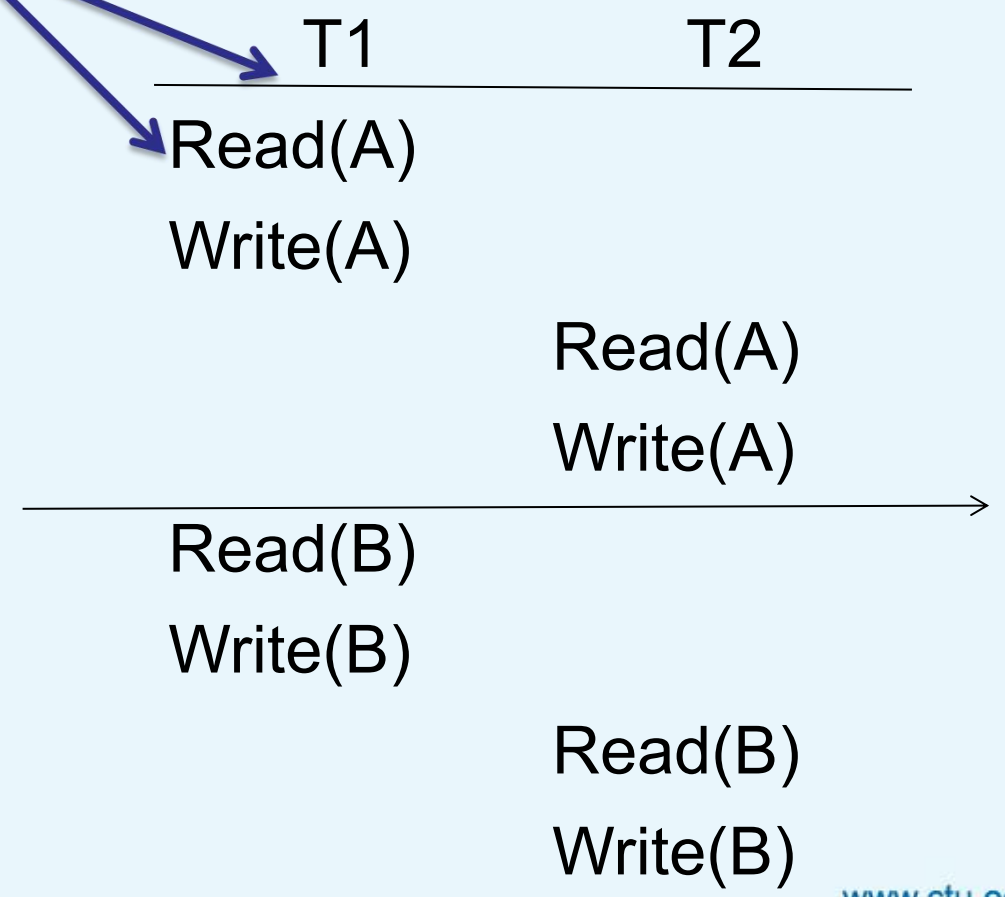
- Chỉ quan tâm thao tác Read và Write trên các dữ liệu
- Giữa Read(X) và Write(X) sẽ có 1 dãy thao tác tùy ý trên bản sao của hạng mục dữ liệu X trong bộ nhớ đệm.

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
	W(B)
Schedule 3	

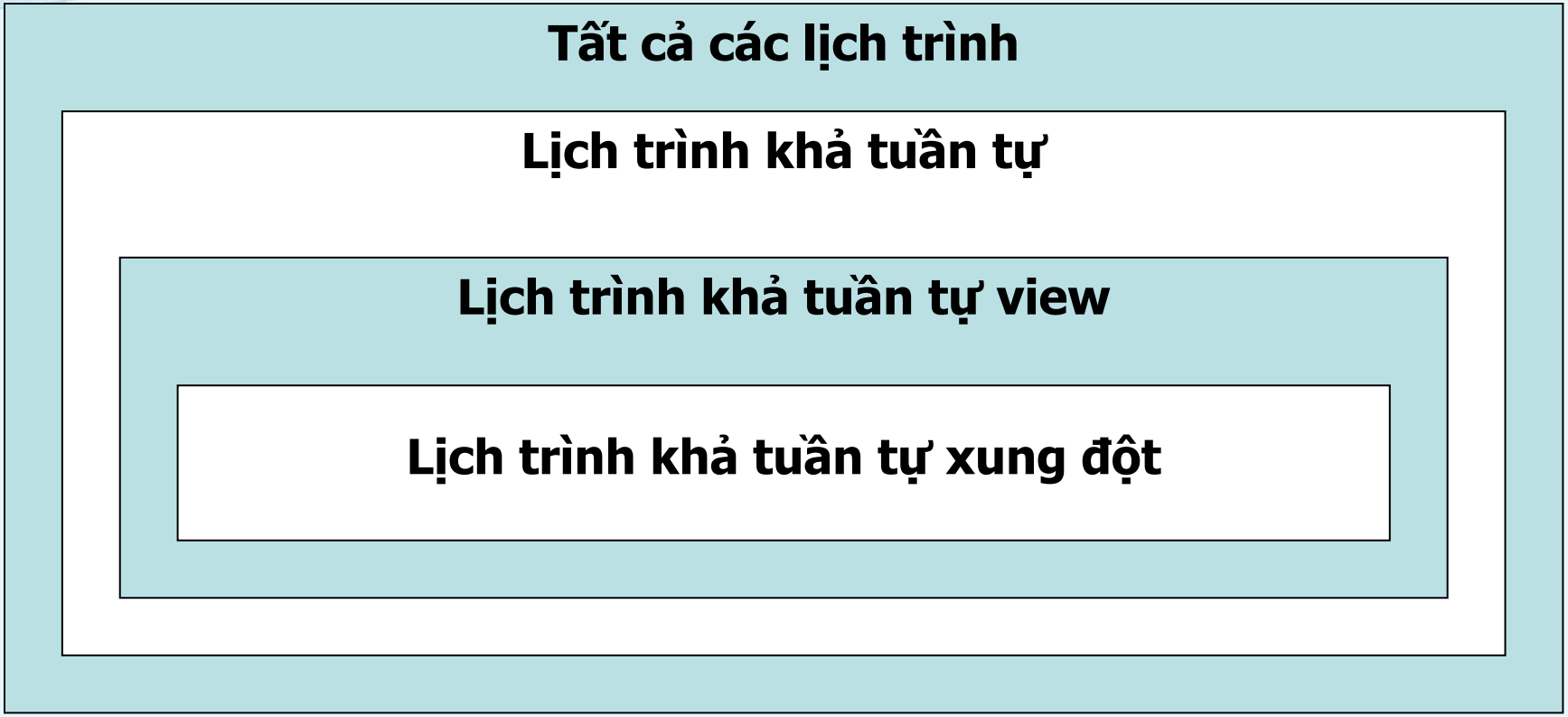
Lịch trình chỉ bao gồm các chỉ thị Read và Write được gọi là lịch trình viết dưới dạng thỏa thuận

➢ Biểu diễn lịch trình

- S_1: $R_1(A)$ $W_1(A)$ $R_2(A)$ $W_2(A)$ $R_1(B)$ $W_1(B)$ $R_2(B)$ $W_2(B)$



➢ Quan hệ giữa các tính chất khả tuần tự



Nếu các GD là nhất quán, các lịch trình tuần tự sẽ nhất quán

5. Tính khả tuần tự (tt)

LỊCH TRÌNH TUẦN TỰ

LỊCH TRÌNH KHẢ TUẦN TỰ

➢ Lịch trình khả tuần tự

Khả tuần tự xung đột (conflict serializable)

- Dựa trên ý tưởng hoán vị các chỉ thị không xung đột chuyển 1 lịch trình đồng thời S về 1 lịch trình tuần tự S'. Nếu có 1 cách biến đổi như vậy thì S là 1 LT KTT XĐ

Khả tuần tự view (View serializable)

- Dựa trên ý tưởng lịch đồng thời S và lịch tuần tự S' đọc và ghi những giá trị dữ liệu giống nhau. Nếu có 1 lịch trình S' như vậy thì S là 1 LT KTT View

5.1 Tính khả tuần tự xung đột

- **Chỉ thị xung đột** (conflict instructions): việc thay đổi thứ tự thực thi chỉ thị ảnh hưởng đến kết quả của LT.
- **Tương đương xung đột** (conflict equivalence): 2 lịch trình S cà S' gọi là tương đương xung đột với nhau nếu như chúng ta có thể biến đổi S về S' hoặc từ S' về S bằng cách đảo chỗ các chỉ thị không xung đột

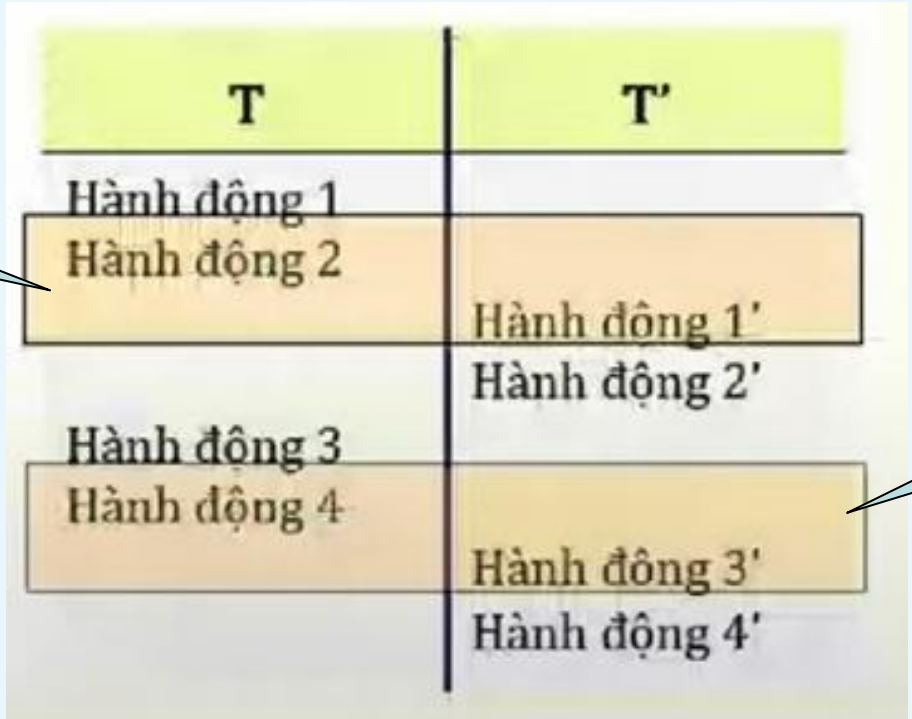
➔ Hai LT tương đương xung đột sẽ cho kết quả giống nhau (cùng trạng thái cuối)

➢ Chỉ thị xung đột

Ý tưởng: Xét 2 chỉ thị liên tiếp nhau của 2 giao dịch khác nhau trong 1 lịch trình, khi 2 chỉ thị thực hiện đảo thứ tự:

Không thay đổi kết quả

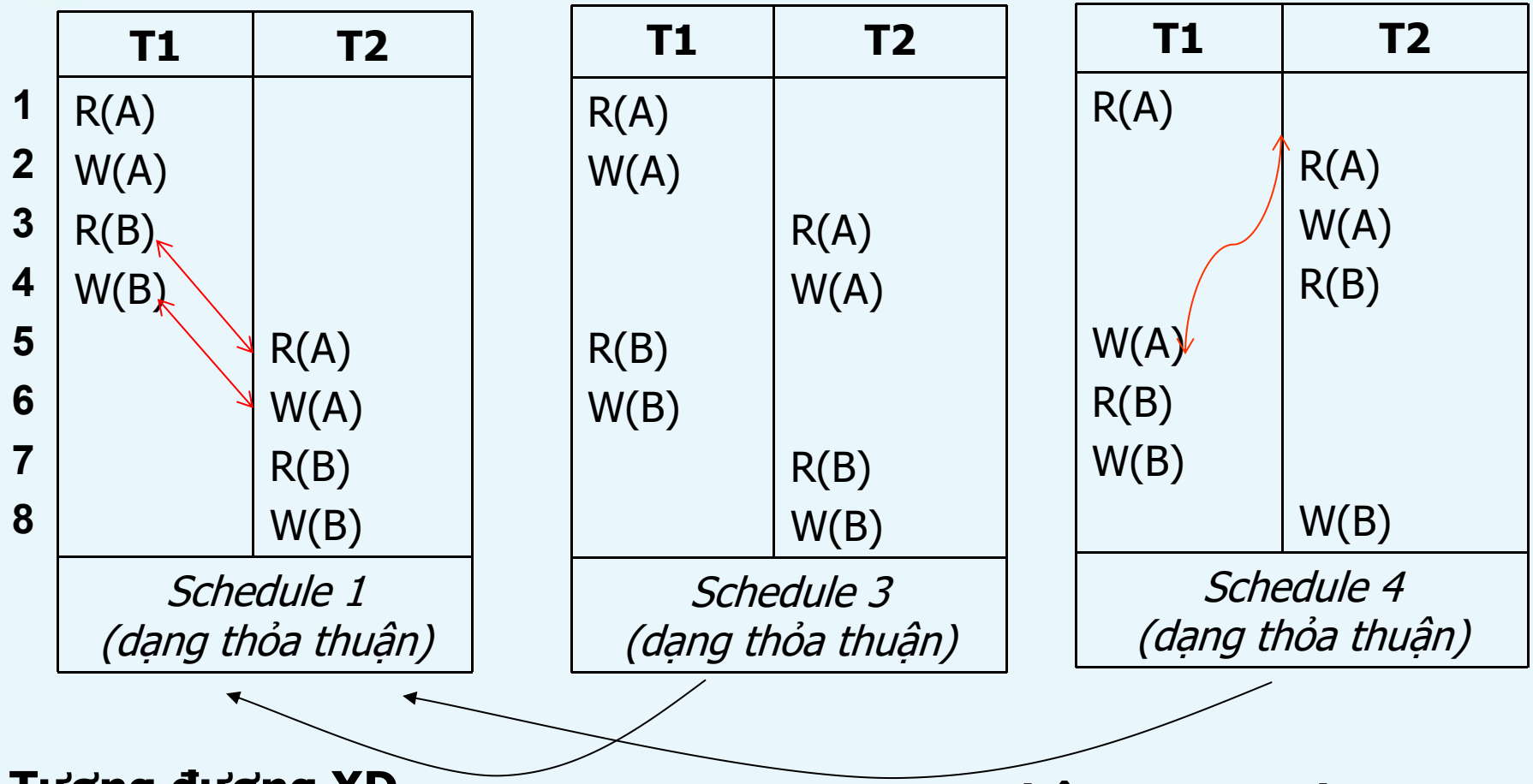
2 chỉ thị không xung đột



Thay đổi kết quả

2 chỉ thị xung đột

Tương đương xung đột



➢ Khả tuần tự xung đột

- Một lịch trình S được gọi là khả tuần tự xung đột (conflict serializable) nếu như S tương đương xung đột với 1 lịch trình tuần tự S'.
- Khi đó ta có thể thay đổi thứ tự các chỉ thị không xung đột trong S cho đến khi ta được một lịch trình tuần tự S' tương đương xung đột S.

➔ **LT khả tuần tự xung đột là 1 LT nhất quán**

➢ Khả tuần tự xung đột (tt)

- 2 chỉ thị trong 1 lịch trình gọi là xung đột khi nó thỏa 3 điều kiện sau:
  - 1. Thuộc về 2 giao dịch khác nhau
  - 2. Thực hiện trên cùng dữ liệu X
  - 3. Ít nhất 1 chỉ thị là Write(X)
- 2 hành động xung đột thì không thể đảo thứ tự chúng trong một thao tác

Ví dụ 1: Xét các cặp xung đột

S_1: $R_1(A)$ $W_1(A)$ $R_2(A)$ $W_2(A)$ $R_1(B)$ $W_1(B)$ $R_2(B)$ $W_2(B)$

- Cặp **$R_1(A)$, $W_2(A)$** là xung đột vì chúng thuộc về hai giao dịch khác nhau trên cùng dữ liệu A và một trong số chúng là lệnh ghi
- Các cặp **$W_1(A)$, $W_2(A)$** và **$W_1(A)$, $R_2(A)$** cũng là xung đột
- Mặc khác, cặp **$R_1(A)$, $W_2(B)$** không xung đột vì chúng hoạt động trên 2 dữ iệu khác nhau
- Tương tự, cặp **$W_1(A)$, $W_2(B)$** là không xung đột

Ví dụ 2: Liệt kê các cặp xung đột

S_2:	T1	T2
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Write(A)
5	Read(B)	
6	Write(B)	
7		Read(B)
8		Write(B)

Có 6 cặp xung đột:

- 1 - 4: $R_1(A) - W_2(A)$
- 2 - 3: $W_1(A) - R_2(A)$
- 2 - 4: $W_1(A) - W_2(A)$
- 5 - 8: $R_1(B) - W_2(B)$
- 6 - 7: $W_1(B) - R_2(B)$
- 6 - 8: $W_1(B) - W_2(B)$

❖ Kiểm tra tính khả tuần tự xung đột bằng đồ thị trình tự

❖ Thuật toán kiểm tra tính khả tuần tự của lịch trình

- Input: S {T1, T2, …, Tn}
- Output: S khả tuần tự hay không?
- Thuật toán: G = (N, E)
  - Xây dựng đồ thị phụ thuộc G:
    - Nút (N): là tất cả các giao dịch trong lịch trình S
    - Cung (E): tập hợp các cung đi từ $T_i \rightarrow T_j$

❖ Kiểm tra tính khả tuần tự xung đột bằng đồ thị trình tự (tt)

- Cung $T_i \rightarrow T_j$ khi:
  1. T_i thực hiện Write(X) trước và T_j thực hiện Read(X) sau
  2. T_i thực hiện Read(X) trước và T_j thực hiện Write(X) sau
  3. T_i thực hiện Write(X) trước và T_j thực hiện Write(X) sau

✓ Nếu đồ thị G không có chu trình ➔ Lịch trình S khả tuần tự xung đột

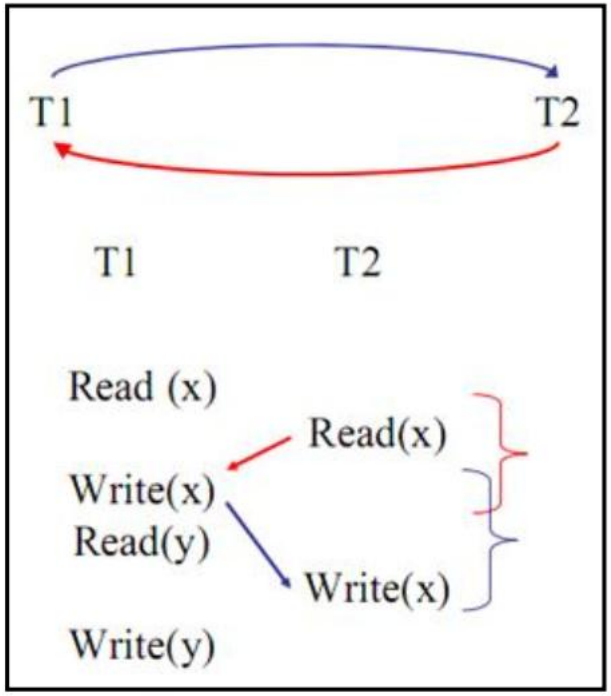
✓ Ngược lại, đồ thị G có chu trình ➔ Lịch trình S không khả tuần tự xung đột

Ví dụ: kiểm tra tính khả tuần tự của lịch trình

- Lịch trình S của 2 GD T1, T2 và đồ thị phụ thuộc G của S

T1	T2
read_item(X)	
X:=X-10	
	read_item(X)
	X:=X-20
write_item(X)	
read_item(Y)	
	write_item(X)
Y:=Y+10	
write_item(Y)	

serializable?



Đồ thị G có chu trình ➔ lịch trình S là **không khả tuần tự xung đột**

✓ Khả tuần tự xung đột

Ví dụ 1: S_1 có khả tuần tự xung đột không?

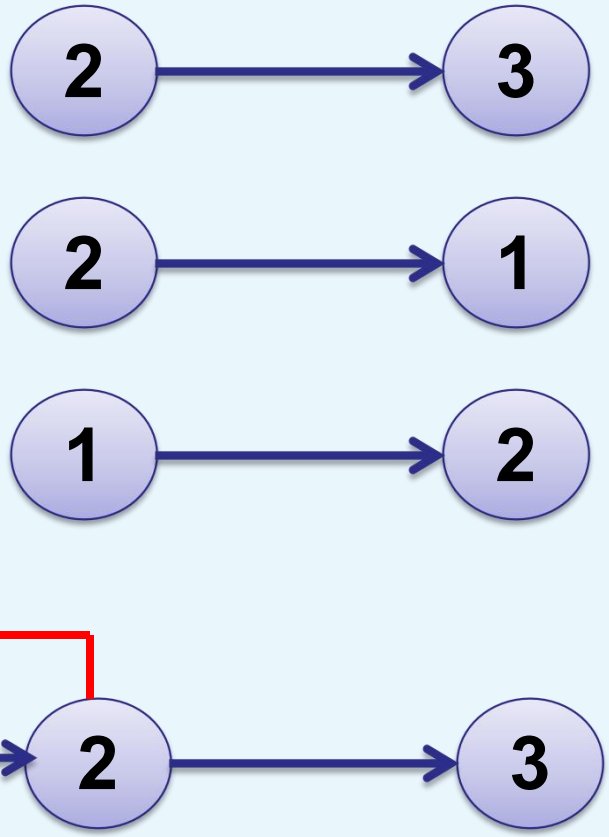
T_1	T_2	T_3
	Read(A)	
Read(B)		
	Write(A)	
		Read(A)
Write(B)		
		Write(A)
	Read(B)	
	Write(B)	

- $P(S_1)$ không có chu trình
- S_1 khả tuần tự xung đột theo thứ tự T_1, T_2, T_3

✓ Không khả tuần tự xung đột

Ví dụ 2: S_2 có khả tuần tự xung đột không?

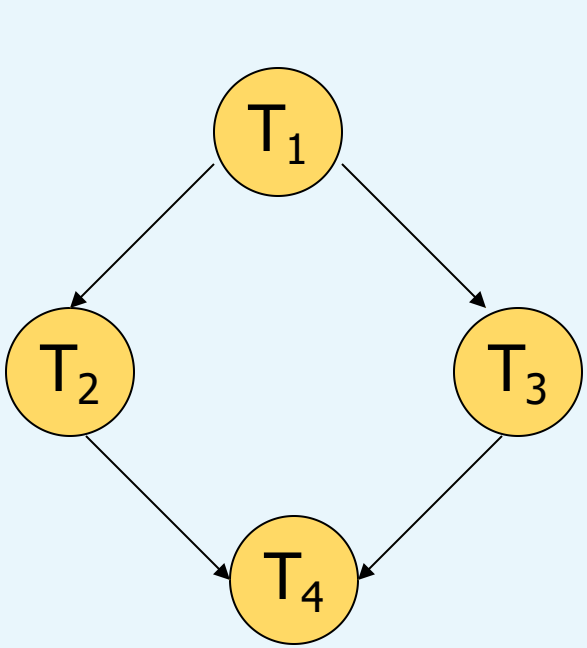
T_1	T_2	T_3
	Read(A)	
Read(B)		
	Write(A)	
	Read(B)	
		Read(A)
Write(B)		
		Write(A)
	Write(B)	

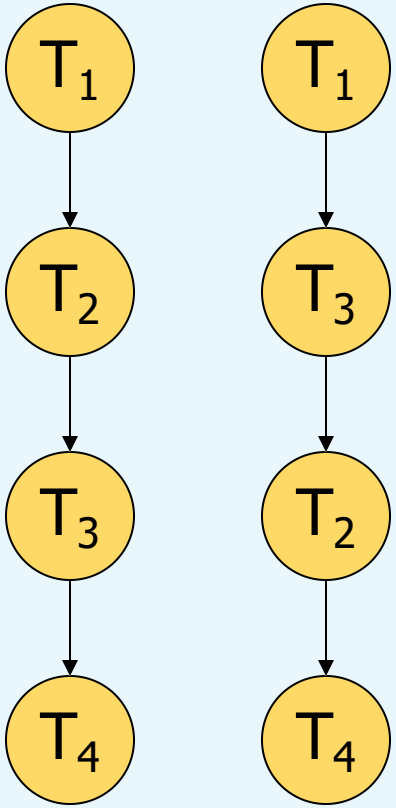


- $P(S_2)$ có chu trình
- S_2 không khả tuần tự xung đột

➢ Thứ tự khả tuần tự xung đột

Đồ thị trình tự không có chu trình





Sắp xếp topo