



# HỆ QUẢN TRỊ CSDL

Số tc: 3 (2 Lec + 1 Lab)

Giảng viên: Lương Thị Thu Phương

Phone: 0935 512 766

Email: [luongthuphuongitdtu@gmail.com](mailto:luongthuphuongitdtu@gmail.com)



# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

## GIAO DỊCH

### 1. Giao dịch là gì?

Giao dịch trong SQL là quá trình thực hiện một chuỗi các thao tác truy vấn SQL (SELECT, INSERT, UPDATE, DELETE) trên CSDL một cách an toàn và đồng nhất.

Mục đích của giao dịch là đảm bảo tính nhất quán và độ tin cậy của cơ sở dữ liệu.

Nguyên lý điều khiển tương tranh: là quá trình điều khiển giúp cho nhiều thao tác diễn ra đồng thời mà không xảy ra tranh chấp.





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

Trong một giao dịch, nếu một câu lệnh không thành công, tất cả các thay đổi được thực hiện trước đó sẽ bị hủy bỏ, và CSDL sẽ trở lại trạng thái ban đầu.

Nếu tất cả các câu lệnh đều thành công, các thay đổi sẽ được lưu trữ vĩnh viễn trong CSDL.

Một giao dịch SQL bao gồm các câu lệnh COMMIT, ROLLBACK để quản lý quá trình thực hiện giao dịch.

Các giao dịch cũng cần được bảo vệ bằng các khóa (lock) để đảm bảo tính nhất quán và độ tin cậy của CSDL.





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

**Ví dụ:**

#### Giao dịch chuyển khoản

*Nếu  $A \geq P$  thì*

*$A := A - P$*

*$B := B + P$*

*Nếu không*

*<Xử lý trường hợp không cho  
phép chuyển khoản>*

*{Không cập nhật tài khoản}*

*Kết thúc nếu*

#### Giao dịch đặt chỗ cho 1 chuyến bay

*Nếu chuyến bay A còn chỗ trống*

*Trừ số ghế trống của chuyến bay A đi 1*

*Cập nhật thông tin khách hàng, mã  
chuyến bay vào CSDL.*

*Kết thúc nếu*



## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

**Thao tác giao dịch:**

**BEGIN TRANSACTION**

**<SQL statements>**

**COMMIT / ROLLBACK**

**END TRANSACTION**





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

Để ghi nhận sự hoàn tất hay không của một thao tác, người ta sẽ sử dụng các lệnh sau:

<i>BEGIN TRANSACTION</i>	<i>Bắt đầu giao tác</i>
<i>COMMIT TRANSACTION</i>	<i>Kết thúc một giao tác thành công</i>
<i>ROLLBACK TRANSACTION</i>	<i>Kết thúc một giao tác không thành công, những thao tác làm ảnh hưởng CSDL trước đó được undo, CSDL được trả về tình trạng trước khi thực hiện giao tác.</i>
<i>END TRANSACTION</i>	<i>Chỉ mạng ý nghĩa hình thức, thường không sử dụng</i>



## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

Ví dụ:

BEGIN TRAN

DELETE FROM DMHANG

WHERE MAHANG = 'MH08'

COMMIT TRAN

MAHANG	TENHANG	DVT	SLNHAP	MANCC
MH01	Bia Larue	thùng	50	CC01
MH02	Sữa Vinamilk	thùng	30	CC02
MH03	Trà Lipton	gram	25	CC03
MH04	Cà fe trung nguyên	kg	60	CC04
MH05	Bia Tiger	thùng	25	CC01
MH06	Sữa ông thọ	thùng	45	CC02
MH07	Sữa cô gái Hà Lan	thùng	50	CC02

BEGIN TRAN

DELETE FROM DMHANG

WHERE MAHANG = 'MH08'

ROLLBACK TRAN

MAHANG	TENHANG	DVT	SLNHAP	MANCC
MH01	Bia Larue	thùng	50	CC01
MH02	Sữa Vinamilk	thùng	30	CC02
MH03	Trà Lipton	gram	25	CC03
MH04	Cà fe trung nguyên	kg	60	CC04
MH05	Bia Tiger	thùng	25	CC01
MH06	Sữa ông thọ	thùng	45	CC02
MH07	Sữa cô gái Hà Lan	thùng	50	CC02
MH08	Cà fe sữa	h?p	35	CC02



## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

**Giả sử muốn chuyển tiền từ TK A sang TK B. Cần thực hiện các bước sau:**

1. Bắt đầu một giao dịch mới bằng cách sử dụng lệnh **BEGIN TRANSACTION:**

**BEGIN TRANSACTION;**

2. Cập nhật số dư của tài khoản A bằng cách trừ số tiền chuyển đi:

**UPDATE accounts**

**SET balance = balance - 1000**

**WHERE account\_number = 'A';**







## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

3. Cập nhật số dư của tài khoản B bằng cách thêm số tiền đã chuyển đến:

UPDATE accounts

SET balance = balance + 1000

WHERE account\_number = 'B';

4. Kiểm tra xem có lỗi nào xảy ra không. Nếu không có lỗi, lưu trữ các thay đổi bằng cách sử dụng lệnh COMMIT:

COMMIT;

5. Nếu có lỗi, hủy bỏ tất cả các thay đổi bằng cách sử dụng lệnh ROLLBACK:

ROLLBACK;



# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

## GIAO DỊCH

### 2 - Các tính chất của giao dịch:

Bất kì 1 giao dịch nào cũng phải thỏa mãn 4 tính chất sau được gọi là tính chất **ACID**.

1. Tính nguyên tử (**A**tomical): Một giao dịch là một đơn vị xử lý không thể chia nhỏ hơn được nữa.
2. Tính nhất quán (**C**onsistency): Giao dịch được gọi là nhất quán khi chuyển CSDL từ tình trạng nhất quán này sang tình trạng nhất quán khác.





# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

## GIAO DỊCH

### 2- Các tính chất của giao dịch:

3. Tính cô lập (**I**ndependence): Các giao dịch xử lý đồng thời phải độc lập với những thay đổi được thực hiện bởi những giao dịch chưa hoàn tất khác.
4. Tính lâu dài (**D**urability): Tất cả những thay đổi trên CSDL mà giao dịch thực hiện cho đến khi nó được xác nhận hoàn tất (COMMIT) thì phải được ghi nhận chắc chắn lên CSDL.





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

### GIAO DỊCH

#### 3. Các vấn đề của việc truy xuất dữ liệu đồng thời

- Mất dữ liệu cập nhật (**Lost update data**):

Ví dụ:

DMHANG (MAHANG, TENHANG, DVT, SLTON)

Giả sử SLTON = 20





# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

## GIAO DỊCH

Giao dịch		Giá trị		
T1 (nhập hàng)	T2 (bán hàng)	x1	x2	SLTON
<i>Begin Transaction</i>	<i>Begin Transaction</i>			20
Read (SLTon) $\rightarrow$ x1		<b>20</b>		20
	Read (SLTon) $\rightarrow$ x2	20	<b>20</b>	20
x1 + 10 $\rightarrow$ x1		<b>30</b>	20	20
	x2 - 3 $\rightarrow$ x2	30	<b>17</b>	20
Write x1 $\rightarrow$ SLTon		30	17	<b>30</b>
	Write x2 $\rightarrow$ SLTon	30	17	<b>17</b>
<i>Commit T1</i>	<i>Commit T2</i>			17

**Nhận xét:** Thao tác cập nhật của giao dịch T1 coi như bị huỷ bỏ, không được cập nhật (do những thay đổi của giao tác khác ghi đè lên)



# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

## 1- GIAO DỊCH

### 3 - Các vấn đề của việc truy xuất dữ liệu đồng thời

- Đọc dữ liệu chưa commit (**Uncommit data**):

Ví dụ:

DMHANG (MAHANG, TENHANG, DVT, SLTON)

Giả sử SLTon = 20.





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

Giao dịch		Giá trị		
T1 (nhập hàng)	T2 (bán hàng)	x1	X2	SLTon
<i>Begin Transaction</i>				20
Read (SLTon) $\rightarrow$ x1		<b>20</b>		20
$x1 + 100 \rightarrow x1$		<b>120</b>		20
Write x1 $\rightarrow$ SLTon		120		<b>120</b>
	<i>Begin Transaction</i>	120		120
	Read (SLTon) $\rightarrow$ x2	120	<b>120</b>	120
	$x2 - 5 \rightarrow x2$	120	<b>115</b>	120
	Write x2 $\rightarrow$ SLTon	120	115	<b>115</b>
	<i>Commit T2</i>	120	115	115
<i>Rollback T1</i>				<b>20</b>

**Nhận xét:** Số lượng tồn cuối cùng là sai vì T2 đọc dữ liệu số lượng tồn chưa được commit, T1 Rollback sau khi T2 đọc dữ liệu.



# Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

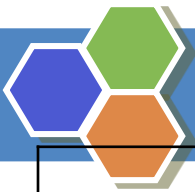
## 1- GIAO DỊCH

### 3- Các vấn đề của việc truy xuất dữ liệu đồng thời

- Thao tác đọc không thể lặp lại (**Unrepeatable data**):







## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

Giao dịch		Giá trị
T1	T2	
<i>Begin Transaction</i>		
Tong = 0		Tong = 0
Read (SLA) $\rightarrow$ x1		x1 = 10
Read (SLB) $\rightarrow$ x2		x2 = 15
	<i>Begin Transaction</i>	
	Read (SLA) $\rightarrow$ x3	x3 = 10
	x3 - 5 $\rightarrow$ x3	x3 = 5
	Write x3 $\rightarrow$ SLA	SLA = 5
	Read (SLC) $\rightarrow$ x4	x4 = 27
Tong = x1 + x2		Tong = 25
	Write x4 $\rightarrow$ SLC	SLC = 27
Read (SLC) $\rightarrow$ x5		x5 = 27
Tong = Tong + x5		Tong = 52
	<i>Commit T2</i>	
<i>Commit T1</i>		

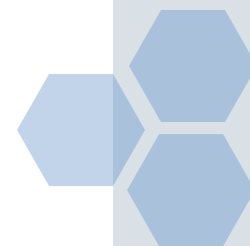
SLA: 10

SLB: 15

SLC: 27

===

Tong 52





## Chương 5: ĐIỀU KHIỂN TƯƠNG TRANH

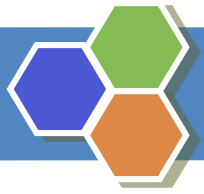
**Nhận xét:** Tổng cuối cùng là sai vì giá trị  $Tong = x1 + x2$  ( $x1$  chính là số lượng tồn A) đã đọc số lượng tồn A trước khi được cập nhật bởi giao dịch khác T2, nên Tong cuối cùng mà giao dịch T1 cập nhật là sai.

→ Để tránh các vấn đề nêu trên, ta cần phải điều khiển tương tranh của các giao dịch truy xuất đồng thời.

**Có 3 cách để điều khiển tương tranh được trình bày:**

1. Điều khiển tương tranh dựa vào tính khả tuần tự
2. Điều khiển tương tranh bằng cơ chế nhãn thời gian
3. Điều khiển tương tranh dựa vào cơ chế khoá (lock)





# BÀI TẬP CÁ NHÂN

1. Nêu các vấn đề truy xuất đồng thời?
2. Cho ví dụ minh họa vấn đề mất dữ liệu cập nhật?
3. Cho ví dụ minh họa vấn đề đọc dữ liệu chưa commit?
4. Thực hiện giao dịch đặt vé máy bay online?

*(Chú ý: Thiết kế Database giả lập cho giao dịch trên)*

