

Tổng quan về Giao tác.

1. Khái niệm giao tác trong CSDL.

- Nó là một dãy các thao tác đọc và ghi có truy vấn hoặc làm thay đổi cơ sở dữ liệu, thao tác được xem như một câu lệnh. Vậy có thể xem nó như một chương trình, hay một phần của chương trình.
- Nếu chỉ có đọc dữ liệu không thì không được coi là giao tác bởi nó không ảnh hưởng đến ràng buộc về dữ liệu
- các thao tác này có thể là một câu lệnh SQL, hay một đoạn mã của chương trình (C++, java)
- Một chương trình ứng dụng sẽ gồm nhiều giao tác + những chương trình không truy vấn đến cơ sở dữ liệu.
- Ví dụ : Một giao tác có truy vấn và làm thay đổi giá của một sản phẩm
`read(product.id = x, price)`
`new_price = price * 1.1`
`write(product.id = x, new_price)`

giao tác trên đọc một sản phẩm có id là x, sau đó tiến hành tăng giá và ghi lại giá trị price vào sản phẩm đấy.

hay đây là một câu lệnh update, một giao tác có vẻ là hoàn chỉnh hơn với Begin end là ranh giới của giao tác

```
Begin transaction BUDGET_UPDATE
begin
UPDATE PROJ
SET BUTGET = BUTGET * 1.1
WHERE PNAME = "CAD/CAM"
end
```

2. Hoạt động của một giao tác trong CSDL

- Một giao tác sẽ đọc/ghi một đơn vị dữ liệu (mục dữ liệu | item) của CSDL thông qua một bộ đệm của giao dịch (vùng làm việc riêng - private work space). Mỗi giao tác có thể thực hiện các thao tác tính toán dẫn đến thay đổi giá trị của dữ liệu, nhưng những thay đổi này sẽ không tác động ngay vào CSDL cho đến khi các giá trị mới được ghi

vào CSDL nhờ một **thủ tục hợp thức (committed)** khi kết thúc giao dịch. Giao dịch chỉ đọc sẽ không làm thay đổi CSDL.

- Nếu giao tác có thể hoàn thành tác vụ của nó thì gọi là nó có ủy thác (**commit** là điểm cuối)
- Nếu có một tác động nào đấy làm gián đoạn giao tác , giao tác sẽ bị hủy bỏ hoàn toàn , các thao tác và CSDL sẽ trở về trạng thái ban đầu, gọi là **abort**
bởi ta phải hiểu nếu như không hủy bỏ đi thì nó sẽ ảnh hưởng đến giao tác khác

3.Các loại giao tác.

a. Giao tác phẳng .

- là giao tác chỉ có điểm khởi đầu và kết thúc duy nhất.(begin-end)

b. Giao tác lồng .

- giao tác lồng : bên trong một cặp begin - end lồng nhiều cặp begin-end khác.
- như vậy thì giao tác lồng biểu đạt một giao tác mà gồm nhiều giao tác khác bên trong , mục đích để thực hiện nhiều giao tác trong 1 giao tác

Begin_transaction Reservation

begin

Begin_transaction Airline

...

Begin_transaction Hotel

...

end. { Hotel }

.....

end. { Airline }

Begin_transaction Car

...

end. { Car }

end.

đây là ví dụ về việc giao tác đặt ô tô , đặt khách sạn được nằm trong việc đặt vé máy bay

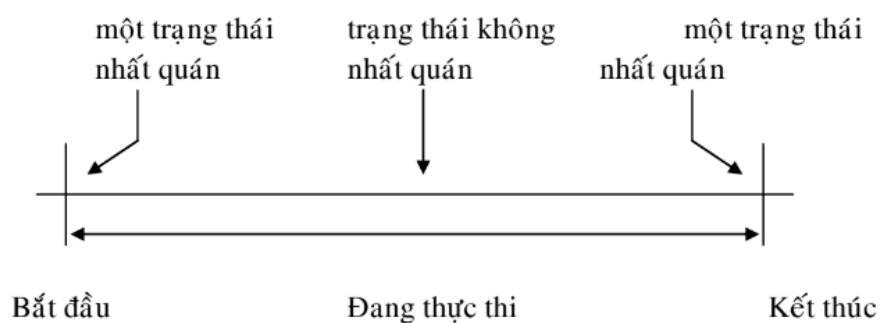
thường là nó phù hợp với những ứng dụng phức tạp

3. Tính chất giao tác . (ACID) để tìm hiểu chi tiết hơn ta có các tính chất của giao tác

1. **A- Nguyên tử:** Mọi thao tác trong giao tác khi thực hiện , một là chúng cùng được thực hiện , hai là không thao tác nào được thực hiện . Nó không có tính phân chia . Tức là nếu nó bị ngắt quãng , hệ thống sẽ khôi phục lại từ đầu, kiểu được ăn cả ngã về không. Như mình đã nói ở phần trước.

2. **C- Nhất quán:** Mọi giao tác phải đảm bảo chuyển CSDL từ trạng thái nhất quán này sang trạng thái nhất quán khác

ví dụ trong một giao dịch ngân hàng chẳng hạn : khi mà chuyển tiền từ tài khoản này sang tài khoản kia , thì CSDL sẽ có hai giao tác sẽ có hai lần update , 1 lần là trừ tiền tk chuyển , cộng tiền tk nhận , làm sao tổng số tiền giữa hai tài khoản luôn không đổi trước và sau khi thực hiện giao tác , đó chính là việc chuyển từ trạng thái nhất quán này sang trạng thái nhất quán khác , và nếu giao dịch không đảm bảo được sự nhất quán về dữ liệu (ví dụ chuyển quá số tiền còn dư trong tài khoản , trong khi tài khoản không thể âm , tổng số tiền giữa hai tài khoản sẽ lớn hơn làm mất tính nhất quán và ràng buộc về dữ liệu) và nếu như không có giao tác ấy , thì giao dịch vẫn sẽ được thực hiện thành công tại phía nhận , do đó trong trường hợp này ta phải dùng đến giao tác để quản lý các giao dịch để đảm bảo CSDL nhất quán dù nhiều truy xuất gặp lỗi



tức là nó có thể gặp lỗi trong giao dịch (lúc đó toàn bộ giao dịch ở các phía sẽ bị hủy bỏ và đưa về trạng thái nhất quán) , hoặc không gặp lỗi nhưng do băng thông hẹp hay gì đấy làm chậm giao dịch (bên A trừ tiền nhưng bên B chưa nhận tiền) thì lúc đó sẽ ở trạng thái chưa nhất quán , nhưng sau cùng khi kết thúc xong nó vẫn ở trạng thái nhất quán

3.1- Tính cô lập

- Các giao tác phải được thực hiện một cách độc lập với nhau.
Nói một cách khác những tác động của giao tác này sẽ không thể thấy được đối với những giao tác khác, khi giao tác này chưa thực hiện xong hoàn toàn

Read(x)	T2:	Read(x)
$x \leftarrow x + 1$		$x \leftarrow x + 1$
Write(x)		Write(x)

Ví dụ :

$x=50$

T ₁ :	Read(x)
T ₁ :	$x \leftarrow x + 1$
T ₁ :	Write(x)
T ₁ :	Commit
T ₂ :	Read(x)
T ₂ :	$x \leftarrow x + 1$
T ₂ :	Write(x)
T ₂ :	Commit

trong trường hợp trên thì 2 giao tác thực hiện lần lượt nên không vấn đề gì

T ₁ :	Read(x)
T ₁ :	$x \leftarrow x + 1$
T ₂ :	Read(x)
T ₁ :	Write(x)
T ₂ :	$x \leftarrow x + 1$
T ₂ :	Write(x)
T ₁ :	Commit
T ₂ :	Commit

T₂ Commit

Trong trường hợp này, giao tác T₂ đọc được giá trị của x là 50. Giá trị này không đúng bởi vì T₂ đọc x trong khi giá trị của nó đang được thay đổi từ 50 thành 51. Hơn nữa giá trị của x sẽ là 51 vào lúc kết thúc các giao tác T₁ và T₂ bởi vì hành động ghi của T₂ sẽ đề lên kết quả ghi của T₁.

4. D - Tính bền vững .

- Những thay đổi của CSDL do tác động của một giao tác thành công là bền vững, không bị mất đi.

Tức là sau khi mà giao tác thực hiện thành công (commit) dù hệ thống có xảy ra lỗi hay như thế nào đi chăng nữa thì kết quả của nó cũng không bị xóa đi , vậy nên nó sẽ ủy thác thành công trước khi thông báo tới người dùng , ví dụ giao dịch thành công chẳng hạn , dù hệ thống có lỗi trước khi nó thông báo tới người dùng thì nó đã commit thành công ,.