An toàn và toàn vẹn dữ liệu

Nguyễn Hồng Phương
phuongnh@soict.hust.edu.vn
http://is.hust.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Nội dung

1. Đặt vấn đề
2. An toàn dữ liệu
3. Toàn vẹn dữ liệu
4. Điều khiển tương tranh

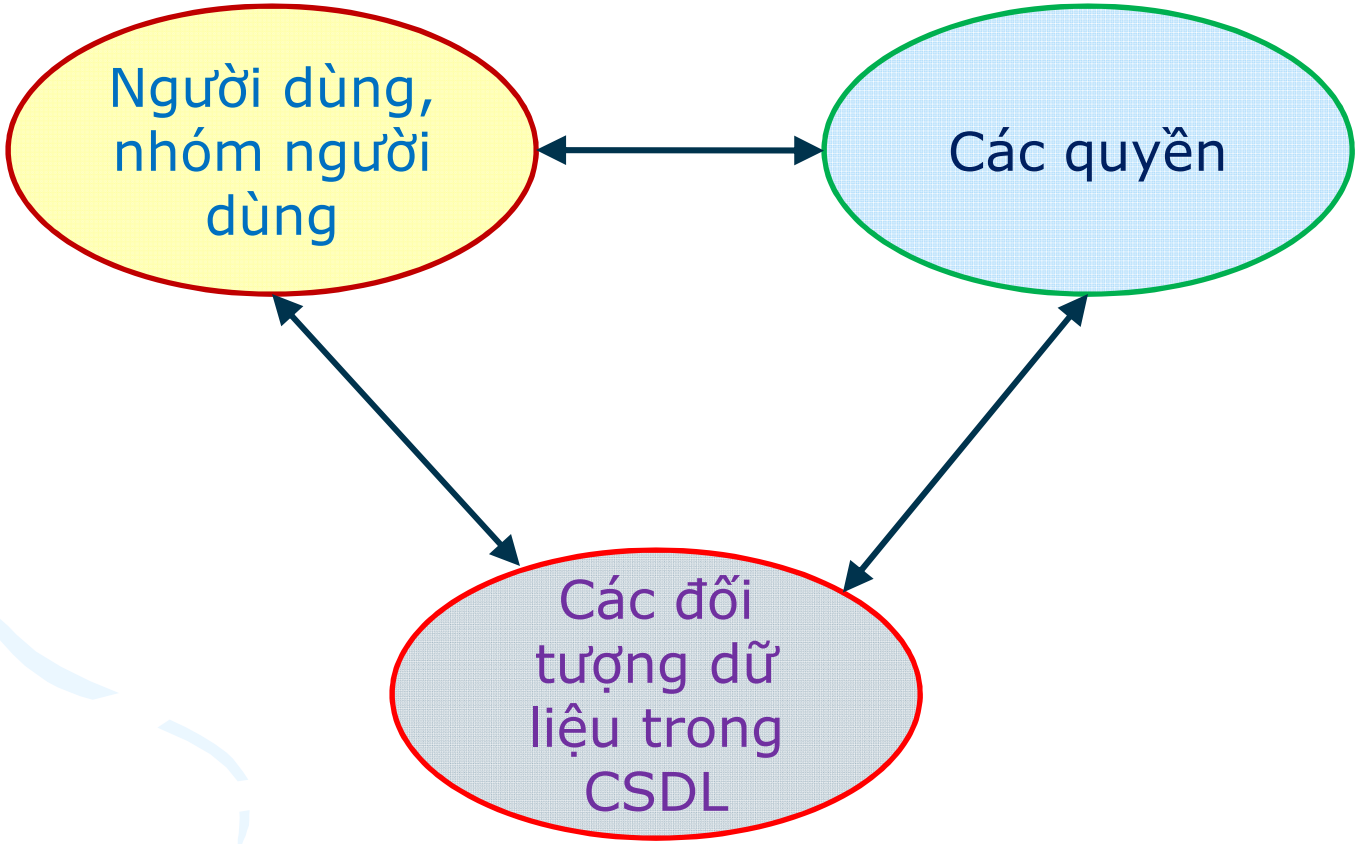
1. Đặt vấn đề

- Một số yêu cầu đối với thiết kế, cài đặt và quản trị CSDL:
  - Đảm bảo tính an toàn của dữ liệu
    - Tránh truy nhập không hợp lệ từ phía người dùng: phân quyền, xác minh và kiểm tra quyền hạn người sử dụng.
  - Đảm bảo tính đúng đắn của dữ liệu
    - Tránh sai sót khi cập nhật dữ liệu: định nghĩa và kiểm tra các ràng buộc dữ liệu.
    - Tránh sai sót trong quá trình thao tác với dữ liệu: kiểm tra tính toàn vẹn của các thao tác với dữ liệu.

2. An toàn dữ liệu

- **Định nghĩa:** Tính an toàn dữ liệu là sự bảo vệ dữ liệu trong CSDL chống lại những truy nhập, sửa đổi hay phá hủy bất hợp pháp.
- Người sử dụng hợp pháp là những người sử dụng được cấp phép, ủy quyền.
- Để đảm bảo tính an toàn cho CSDL, cần có một cơ chế để quản lý người dùng hợp lý.
- Những nhóm người dùng khác nhau trong hệ CSDL có quyền sử dụng khác nhau đối với các đối tượng dữ liệu trong CSDL.

Trục tam giác



2.1. Các quyền truy nhập của người sử dụng

- Đối với người khai thác CSDL:
  - Quyền đọc dữ liệu: được phép đọc một phần hay toàn bộ dữ liệu trong CSDL.
  - Quyền cập nhật dữ liệu: được phép sửa đổi một số giá trị nhưng không được xóa dữ liệu trong CSDL.
  - Quyền xóa dữ liệu: được phép xóa dữ liệu trong CSDL.
  - Quyền bổ sung dữ liệu: được phép thêm dữ liệu mới vào trong CSDL nhưng không được phép thay đổi dữ liệu

2.1. Các quyền truy nhập của người sử dụng

- Đối với người quản trị CSDL:
  - Quyền tạo chỉ dẫn trên các quan hệ trong CSDL
  - Quyền thay đổi sơ đồ cơ sở dữ liệu: thêm hay xóa các thuộc tính của các quan hệ trong CSDL
  - Quyền loại bỏ quan hệ trong CSDL
  - Quyền quản lý tài nguyên: được phép thêm các quan hệ mới vào CSDL

2.2. Người dùng

- Một người dùng cụ thể (user)
  - ví dụ: phuongnh, postgres,…
- Một nhóm người dùng (group)
  - nhóm có các thành viên (member)
  - ví dụ: group_h3t có phuongnh, lamdb, trinhvt,….
- Một số hệ quản trị không phân biệt 2 khái niệm trên, mà gọi chung là vai trò (role)
  - ví dụ: trong PostGreSQL, role có thể là user hoặc group (nếu có member)

2.3. Các đối tượng dữ liệu

- Tables
- Views

Trách nhiệm của người quản trị hệ thống

- Để có thể phân biệt được người sử dụng trong hệ CSDL, người quản trị hệ thống phải có trách nhiệm:
  - Xác định các quyền cụ thể mà mỗi người sử dụng hay một nhóm người sử dụng được phép thực hiện, xác định vai trò và trách nhiệm của mỗi người sử dụng. Điều này được gọi chung là Phân quyền người sử dụng.
  - Cung cấp một phương tiện cho người sử dụng để hệ thống có thể nhận biết được người sử dụng đó hay còn gọi là Xác minh người sử dụng.

Xác minh người sử dụng

- Để xác minh được người sử dụng, người ta có thể dùng các kỹ thuật sau:
  - Kỹ thuật dùng tài khoản có tên và mật khẩu, mật khẩu cũng được bảo vệ bởi hệ thống.
  - Kỹ thuật sử dụng các hàm kiểm tra người sử dụng: Hệ thống đưa cho người sử dụng một số ngẫu nhiên x, người sử dụng dùng một hàm F tính nhẩm kết quả và đưa kết quả y = F(x) vào hệ thống. Trong lúc đó, hệ thống cũng tính toán và so sánh kết quả với y. Người sử dụng hợp pháp là người biết hàm biến đổi F và đưa vào giá trị y đúng.
  - Kỹ thuật dùng thẻ điện tử, thẻ thông minh.
  - Kỹ thuật sử dụng nhận dạng tiếng nói, vân tay v..v.

Kiểm tra quyền truy nhập của người sử dụng

- Mỗi người sử dụng sẽ có một bộ hồ sơ do người quản trị thiết lập và được hệ thống quản lý, trong hồ sơ đó sẽ có chi tiết về các thao tác người sử dụng được phép thực hiện:
  - Phân quyền người sử dụng: Người quản trị hệ thống phải có trách nhiệm xác định khung nhìn để kiểm soát xem mỗi người sử dụng chỉ được truy nhập phần dữ liệu nào trong CSDL và có được các quyền nào trong số các quyền đọc, thêm, xóa, sửa đổi.
  - Xác định và kiểm soát sự lưu chuyển dữ liệu: Hệ thống phải bảo trì danh sách các quyền một cách chặt chẽ vì người sử dụng có thể được quyền lan truyền các quyền cho người sử dụng khác.

Các câu lệnh an toàn dữ liệu trong SQL

- Câu lệnh tạo khung nhìn
- Câu lệnh phân quyền cho người sử dụng
- Câu lệnh thu hồi quyền của người sử dụng

Câu lệnh tạo khung nhìn

- CREATE VIEW <Tên khung nhìn> [(d/s cột)] AS <Câu truy vấn>
- Danh sách các cột trong khung nhìn là phần không bắt buộc. Trong trường hợp người sử dụng muốn đặt tên khác cho các cột xuất hiện trong khung nhìn thì người sử dụng có thể chỉ ra tên các cột, dữ liệu trên cột thì tương ứng với các cột trong mệnh đề Select của câu truy vấn.

Ví dụ câu lệnh tạo khung nhìn

- Cho cơ sở dữ liệu gồm 2 quan hệ:

Nhânviên(Id,Họtên,ĐC,Lương,NămBD,Đánhgiá,PhòngCT)
Phòng(PId, Tên, ĐC, Điệnthoại, Trưởngphòng)

- Câu lệnh tạo khung nhìn cho một nhân viên của phòng Khoa Học có thể được định nghĩa như sau:

```
CREATE VIEW NVKH(HọtênNhânviên, Địachỉliênlạc) AS
  SELECT Họtên, Địachỉ
  FROM Nhânviên
  WHERE PhòngCT IN
       (SELECT PId FROM Phòng WHERE Tên ='Khoa Học')
```

Câu lệnh phân quyền cho NSD

- GRANT <D/s thao tác> ON <Đối tượng>
  TO <D/s người dùng> [WITH GRANT OPTION]
- <D/s thao tác>: có thể bao gồm 1 hay nhiều thao tác được liệt kê dưới đây:
  - Insert: chèn dữ liệu vào trong CSDL có sẵn nhưng không được thay đổi bất kỳ mục dữ liệu nào trong CSDL
  - Update: sửa đổi dữ liệu nhưng không được xóa dữ liệu
  - Delete: xóa dữ liệu trong CSDL
  - Select : tìm kiếm
  - Create: tạo lập các quan hệ mới
  - Alter: Thay đổi cấu trúc của quan hệ
  - Drop: Loại bỏ quan hệ
  - Read/Write: Đọc và Ghi

Câu lệnh phân quyền cho NSD (tiếp)

- <Đối tượng>: bảng hoặc khung nhìn
- <D/s người dùng>: Một người hay một nhóm hay một danh sách người sử dụng. Từ khóa public được dùng thay thế cho mọi người sử dụng
- [With Grant Option] Nếu dùng từ khóa này trong câu lệnh phân quyền thì người dùng xuất hiện trong <D/s người dùng> có quyền được lan truyền các quyền vừa được tuyên bố cho những người dùng khác

Ví dụ câu lệnh phân quyền cho NSD

- Trao quyền đọc, ghi, tìm kiếm, sửa đổi dữ liệu cho nhân viên tên Hoa của phòng Khoa học trên khung nhìn vừa tạo lập trong phần trước

  GRANT read, write, select, update ON NVKH TO Hoa;
- Trao quyền cho trưởng phòng Khoa học – ông HungNC

  GRANT read, write, select, update, delete ON NVKH TO HungNC WITH GRANT OPTION;

Câu lệnh thu hồi quyền của NSD

- REVOKE <D/s thao tác> ON <Đối tượng> FROM <D/s người dùng> [RESTRICT/CASCADE]
- <D/s thao tác>, <Đối tượng>, <D/s người dùng> giống như đối với câu lệnh GRANT.
- Phần [RESTRICT/CASCADE] là chỉ ra cơ chế thu hồi với các quyền đã được người dùng trong <D/s người dùng> lan truyền

Câu lệnh thu hồi quyền của NSD (tiếp)

- Nếu Restrict thì có nghĩa là chỉ hủy bỏ quyền của những người có trong danh sách, quyền đã được lan truyền cho người khác không bị thu hồi.
- Nếu dùng Cascade thì hủy bỏ quyền của người trong <D/s người dùng>, đồng thời kéo theo hủy bỏ quyền mà người dùng đó đã luân chuyển cho những người khác.
- Ví dụ:

  REVOKE update, delete ON NVKH FROM HungNC CASCADE

3. Toàn vẹn dữ liệu

- Định nghĩa: Tính toàn vẹn dữ liệu là sự bảo vệ dữ liệu trong CSDL chống lại những sự sửa đổi, phá hủy vô căn cứ để đảm bảo tính đúng đắn và chính xác của dữ liệu.
- Các thao tác có thể ảnh hưởng đến tính đúng đắn của CSDL là thêm, xóa, sửa đổi.

- Để đảm bảo tính toàn vẹn dữ liệu, cần phải chỉ ra và duy trì những ràng buộc toàn vẹn liên kết với mỗi quan hệ. Các ràng buộc toàn vẹn cung cấp 1 phương tiện để đảm bảo rằng các thao tác được thực hiện bởi những người sử dụng hợp pháp không làm mất đi tính đúng đắn của CSDL.
- Trong hệ thống đa người dùng, để đảm bảo được toàn vẹn dữ liệu, hệ thống còn phải có được một trình điều khiển tương tranh để tránh đụng độ giữa các thao tác được đưa ra bởi những người sử dụng khác nhau tại cùng một thời điểm

Các ràng buộc toàn vẹn trong SQL

- Các ràng buộc về khóa chính, khóa ngoài, kiểm tra miền giá trị sử dụng Check đã được đề cập đến khi nói về câu lệnh tạo bảng trong CSDL.
- Các khẳng định (assertion)
- Các kích hoạt (trigger)

Các khẳng định

- Là một vị từ biểu thị một điều kiện mà CSDL phải luôn luôn thỏa mãn.
- Các khẳng định được tạo ra bằng câu lệnh:

  CREATE ASSERTION <Tên khẳng định> CHECK <Vị từ>

Ví dụ về khẳng định

- Số lượng mặt hàng được cung cấp bởi các hãng có số nhân viên < 50 phải nhỏ hơn 100:

```
CREATE ASSERTION KĐSốlượng CHECK
  NOT EXISTS
  (SELECT * FROM S WHERE
  numofemps < 50 AND sid IN
(SELECT sid FROM SP WHERE quantity
  >= 100))
```

Ví dụ về khẳng định (tiếp)

- Lương của nhân viên không được cao hơn lương người quản lý phòng ban của nhân viên đó.

```
CREATE ASSERTION Salary_Constraint
CHECK (NOT EXISTS
        (SELECT * FROM Employee E,
        Employee M, Department D
        WHERE E.Salary>M.Salary AND
E.Dno=D.Number AND D.MgrSSN=M.SSN))
```

Các kích hoạt (trigger)

- Là một thủ tục lưu trữ hệ thống (stored procedure) đặc biệt, được thực thi một cách tự động khi có sự kiện gây biến đổi dữ liệu như Update, Insert hay Delete
- Được dùng để đảm bảo toàn vẹn dữ liệu hay thực hiện các quy tắc nghiệp vụ nào đó.
- Khi nào sử dụng trigger?
  - khi các biện pháp đảm bảo toàn vẹn dữ liệu khác như Constraint không thể thỏa mãn yêu cầu của ứng dụng

Các kích hoạt (trigger)

- Constraint thuộc loại toàn vẹn dữ liệu khai báo: kiểm tra dữ liệu trước khi cho phép nhận vào bảng
- Trigger thuộc loại toàn vẹn dữ liệu thủ tục nên việc Insert, Update, Delete xảy ra rồi mới kích hoạt trigger.

– Đôi khi, do nhu cầu thay đổi dây chuyền, có thể sử dụng trigger

- Đặc điểm của trigger
  - một trigger có thể làm nhiều công việc, có thể được kích hoạt bởi nhiều sự kiện

Các kích hoạt (trigger)

- trigger không thể được tạo ra trên bảng tạm hoặc bảng hệ thống
- trigger chỉ có thể được kích hoạt tự động bởi các sự kiện mà không thể chạy thủ công được.
- có thể áp dụng trigger cho view
- khi trigger được kích hoạt
  - dữ liệu mới được insert sẽ được chứa trong bảng "inserted"
  - dữ liệu mới được delete sẽ được chứa trong bảng "deleted"
  - đây là hai bảng tạm nằm trên bộ nhớ, và chỉ có giá trị bên trong trigger

Ví dụ về trigger

- Nhânviên(ID,Họtên,Lương,Địachỉ,Ngư ờiquảnlý)
- Một nhân viên bao giờ cũng có lương ít hơn lương người trưởng phòng, điều kiện này phải được kiểm tra khi thêm bộ dữ liệu.

```
CREATE TRIGGER ThemNV ON INSERT Nhânviên
IF Nhânviên.Lương > (SELECT E.Lương FROM
   Nhânviên AS E WHERE E.ID =
   Nhânviên.Ngườiquảnlý)
THEN ABORT;
```

4. Điều khiển tương tranh

- Trong hệ CSDL đa người dùng, hệ thống cần đưa ra giải pháp chống đụng độ giữa các giao dịch (một dãy các thao tác) được đưa ra bởi những người dùng khác nhau để tránh việc một đối tượng dữ liệu nào đó bị làm mất tính đúng đắn trong quá trình cập nhật.

4.1. Giao dịch

- Một giao dịch hình thành nên 1 đơn vị công việc trong 1 DBMS đ/v 1 CSDL, được coi là cố kết và tin cậy độc lập với các giao dịch khác.
- Giao dịch có nhiều bước, và phải được thực hiện một cách trọn vẹn.
- Trạng thái trung gian giữa các bước là ẩn đ/v các giao dịch khác.
- Nếu có sự cố mà giao dịch không thể hoàn thành, thì tất cả các bước không ảnh hưởng lên CSDL.

Ví dụ về giao dịch

- Một CSDL ngân hàng chứa thông tin tài khoản của các khách hàng và tài khoản quỹ của các chi nhánh.
- Giả sử, thực hiện giao dịch chuyển 100 đô-la từ tài khoản của Alice sang tài khoản của Bob:

```
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';
UPDATE branches SET balance = balance - 100.00
    WHERE name = (SELECT branch_name FROM accounts
                  WHERE name = 'Alice');
UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Bob';
UPDATE branches SET balance = balance + 100.00
    WHERE name = (SELECT branch_name FROM accounts
                  WHERE name = 'Bob');
```

4.2. Các tính chất của một giao dịch

- Tính nguyên tố (Atomicy)
  - Hoặc là tất cả các thao tác được thực hiện hoặc là không thao tác nào được thực hiện
  - Đảm bảo tính không thể chia cắt, không thể rút gọn
- Tính nhất quán (Consistency)
  - Dữ liệu nhất quán sau khi giao dịch thực hiện.
- Tính cách ly (Isolation)
  - xác định cách mà những thay đổi bởi một thao tác là ẩn với các thao tác đồng thời khác.

4.2.Các tính chất của một giao dịch

- Tính bền vững (Durability)
  - Đảm bảo giao dịch đã được xác nhận sẽ tồn tại vĩnh cửu.
  - ví dụ, một vị trí chỗ ngồi trên máy bay đã được đặt chỗ thì vị trí đó sẽ vẫn trong trạng thái đã đặt chỗ cho dù hệ thống có vấn đề.
- Tất cả 4 tính chất trên, gọi tắt là ACID

- Các thao tác hình thành nên một giao dịch có thể được nhúng trong một chương trình ứng dụng hoặc được xác định bởi ngôn ngữ cấp cao như SQL.
- Để biết một giao dịch diễn ra, cần xác định tường minh câu lệnh bắt đầu (begin transaction) và kết thúc (end transaction) của giao dịch trong chương trình ứng dụng.

- Giao dịch chỉ đọc (read-only)
  - không cập nhật dữ liệu, mà chỉ lấy ra dữ liệu.
- Để đơn giản, giả sử các thao tác truy cập CSDL trong một giao dịch gồm có:
  - read_item(X): đọc 1 khoản mục CSDL tên là X vào một biến trong chương trình cũng tên là X.
  - write_item(X): ghi giá trị của biến X trong chương trình vào khoản mục CSDL cũng tên là X.

- Đơn vị cơ bản của việc chuyển dữ liệu từ đĩa vào bộ nhớ chính là khối (block).
- Thực hiện lệnh read_item(X) bao gồm các bước sau:
  - Tìm địa chỉ của khối trên đĩa mà chứa khoản mục X.
  - Sao chép khối đó vào bộ đệm trên bộ nhớ chính (nếu trên bộ đệm chưa có).
  - Sao chép khoản mục X từ bộ đệm vào biến chương trình X.

- Thực hiện lệnh write_item(X) bao gồm các bước sau:
  - Tìm địa chỉ khối trên đĩa mà nó chứa khoản mục X.
  - Sao chép khối đó vào bộ đệm trong bộ nhớ chính (nếu bộ đệm chưa có).
  - Sao chép khoản mục X từ biến chương trình tên là X vào vị trí chính xác trên bộ đệm.
  - Lưu trữ khối đã cập nhật này từ bộ đệm lên đĩa (có thể ngay tức thì hoặc lưu trữ sau).

4.3. Tại sao lại phải điều khiển tương tranh?

- Các vấn đề
  - The Lost Update
  - The Temporary Update (Dirty Read)
  - The Incorrect Summary
  - The Unrepeatable Read

The lost update

- Hiện tượng này xuất hiện khi hai giao dịch truy cập vào cùng các khoản mục dữ liệu nhưng thao tác của hai giao dịch lại xen kẽ, làm cho giá trị của các khoản mục dữ liệu không còn đúng nữa.

Time

T_1	T_2
read_item *(X);*	
X:=X-N;	
	read_item *(X);*
	X:=X+M;
write_item *(X);*	
read_item *(Y);*	
	write_item *(X);*
Y:=Y+N;	
write_item *(Y);*	

The Temporary Update

- Xuất hiện khi 1 giao dịch cập nhật 1 khoản mục dữ liệu, nhưng sau đó không hoàn thành các bước tiếp theo => giao dịch không trọn vẹn nên phải roll-back giá trị đã cập nhật về giá trị cũ.
- Khoản mục dữ liệu đã được cập nhật kia lại được sử dụng bởi 1 giao dịch khác trước khi nó được roll-back về giá trị cũ

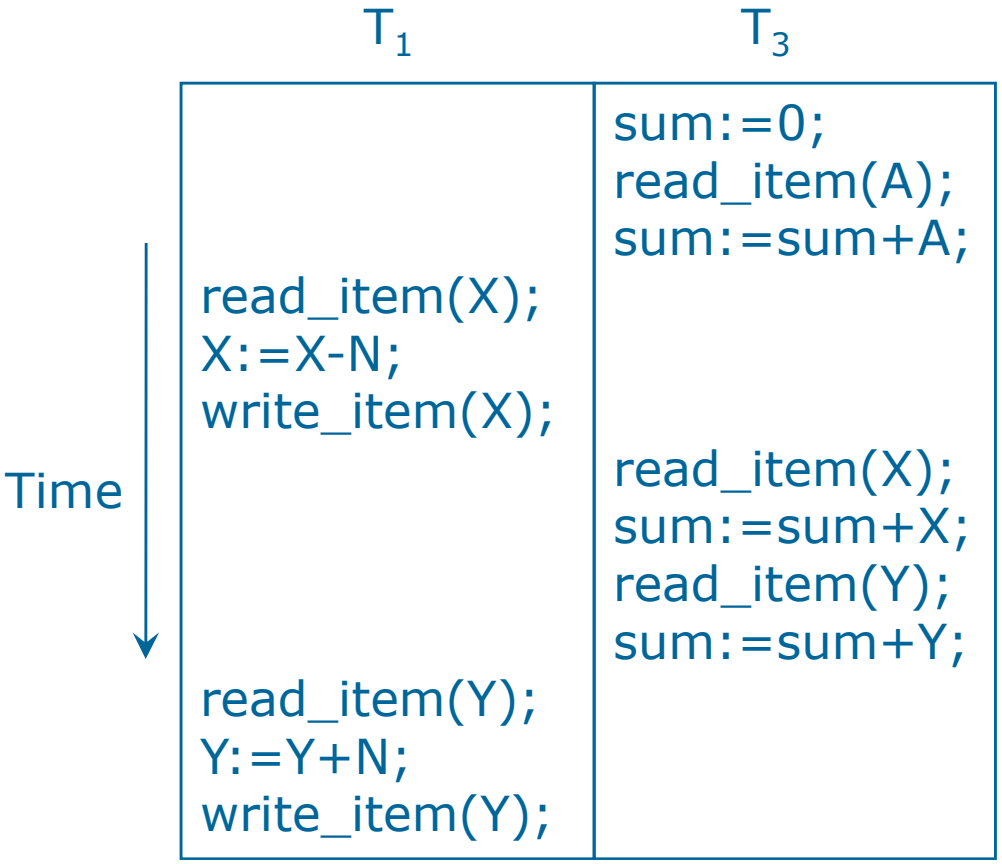
Time

T_1	T_2
read_item *(X);*	
X:=X-N;	
write_item *(X);*	
	read_item *(X);*
	X:=X+M;
	write_item *(X);*
read_item *(Y);*	

Giao dịch *T1 fails và phải thay đổi giá trị của X về giá trị cũ của nó; trong khi đó T2* lại đọc giá trị không đúng tạm thời của X.

The Incorrect Summary

- Một giao dịch tính hàm tích lũy trên các bản ghi đang bị cập nhật bởi 1 giao dịch khác =>hàm tích lũy này có thể tính toán dựa trên một số giá trị đã cập nhật và một số giá trị chưa cập nhật.



T3 đọc X sau khi X đã trừ N và đọc Y trước khi Y cộng N

The Unrepeatable Read

- Một giao dịch T đọc một khoản mục dữ liệu 2 lần. Khoản mục dữ liệu này bị một giao dịch khác thay đổi giữa 2 lần đọc đó.
- Do đó, T nhận các giá trị khác nhau cho 2 lần đọc cùng một khoản mục.

4.4. Các kỹ thuật điều khiển tương tranh

- **Kỹ thuật dùng khóa**: Khi một giao dịch cần dữ liệu nào thì xin hệ điều hành một khóa trên phần dữ liệu đó, các giao dịch khác phải đợi đến khi giải phóng khóa mới được sử dụng phần dữ liệu đó. Có thể người ta sử dụng các loại khóa khác nhau ví dụ như khóa đọc – cho phép nhiều giao dịch đọc cùng 1 lúc, khóa ghi – chỉ 1 giao dịch có được tại một thời điểm.
- **Kỹ thuật gán nhãn thời gian**: Mỗi giao dịch được gán một nhãn T theo thời gian, giao dịch nào cần được ưu tiên thì được gán nhãn thời gian nhỏ hơn và được thực hiện trước. Kỹ thuật này giúp đưa yêu cầu đồng thời về thực hiện tuần tự.

Lời hay ý đẹp

"Khi nói sự thật bạn sẽ không phải nhớ mình vừa nói gì, mà bạn cũng không bao giờ quên những gì mình vừa nói"

S.Raybum