đồng thời vấn đề

Động lực

- Thường RDBMS được sử dụng bởi nhiều người đồng thời (đồng thời kiểm soát)
  - Các ví dụ
    - Ngân hàng trực tuyến
    - Thị trường chứng khoán trực tuyến
    - Bất cứ điều gì trực tuyến !!!
- Thông thường quá trình phải đọc một giá trị (chọn) từ một bảng, vận dụng nó (thay đổi giá trị), và viết nó trở lại cơ sở dữ liệu (cập nhật)

giao dịch

giao dịch

- Giao dịch
  - quá trình liên quan đến truy vấn cơ sở dữ liệu và / hoặc sửa đổi
- Một nhóm các báo cáo (SQL) đóng vai trò như một đơn vị logic thực hiện
- Thông thường với một số đặc tính mạnh mẽ liên quan đến đồng thời
- Hình thành trong SQL từ câu lệnh đơn hoặc kiểm soát lập trình rõ ràng
- Đảm bảo tính nhất quán logic của cơ sở dữ liệu cho các giao dịch có hoạt động mâu thuẫn

tính chất ACID của giao dịch

- số nguyên tử số
  - yêu cầu mỗi giao dịch được coi là một tuyên bố duy nhất thực hiện ("tất cả hoặc không có gì"). Toàn bộ giao dịch thành công, hoặc không ai trong số đó được thực thi, ngay cả sau khi một vụ tai nạn cơ sở dữ liệu.
- Tính nhất quán
  - đảm bảo rằng bất kỳ giao dịch sẽ mang lại cơ sở dữ liệu từ một trạng thái có giá trị khác. Bình thường cũng rất quan trọng ở đây.
- Cô lập
  - đảm bảo rằng việc thực hiện đồng thời các giao dịch dẫn đến một trạng thái hệ thống đó sẽ thu được nếu các giao dịch được thực hiện nối tiếp. Đây là tài sản của RDBMS nới lỏng thường xuyên nhất.
- Độ bền
  - Khi một giao dịch đã được “cam kết” kết quả sẽ ở lại trong cơ sở dữ liệu, ngay cả sau khi một vụ tai nạn cơ sở dữ liệu. Nó thường đòi hỏi cập nhật vào bộ nhớ non-volatile.

COMMIT và ROLLBACK

- COMMIT (Câu lệnh SQL) gây ra một giao dịch để hoàn thành
  - Đó là thay đổi cơ sở dữ liệu hiện nay là vĩnh viễn trong cơ sở dữ liệu
- ROLLBACK (Câu lệnh SQL) cũng khiến giao dịch đến cuối, nhưng bằng cách hủy
  - Không có tác dụng trên cơ sở dữ liệu
  - ROLLBACK sẽ undo tất cả đã thay đổi kể từ lần COMMIT hoặc là ROLLBACK

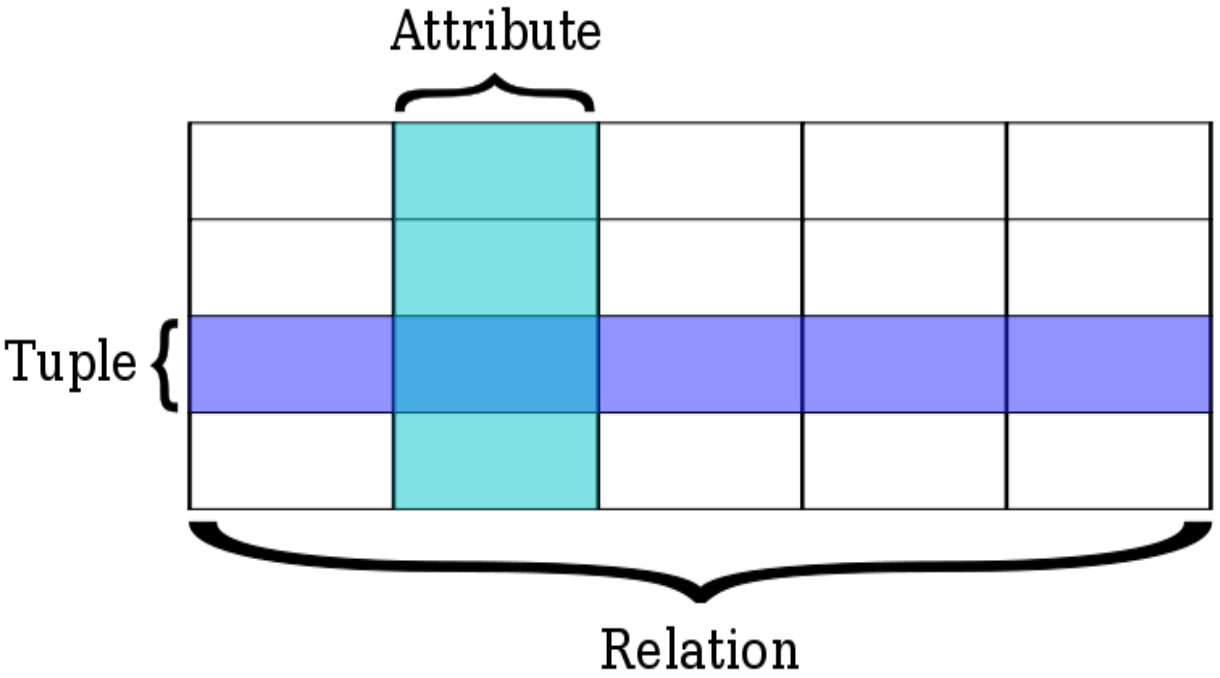
Các ví dụ

- Bạn có thể dùng COMMIT sau khi bất kỳ câu lệnh SQL để đảm bảo rằng cơ sở dữ liệu được ghi vào đĩa (bộ nhớ non-volatile)
- Đôi khi điều này được tự động bật cho câu lệnh SQL (Autocommit)
- Nó có thể được tắt một cách dễ dàng

Thuật ngữ cho Theory

- Cơ sở dữ liệu có thể là mục
  - liên tiếp, một khối đĩa, một thuộc tính, một loạt các tế bào
- read_item (X)
  - Đọc một mục cơ sở dữ liệu X vào một biến
- write_item (X)
  - Ghi vào một mục cơ sở dữ liệu X

Thí dụ



đồng thời vấn đề

- vấn đề cập nhật mất
- cập nhật tạm thời (hoặc bẩn đọc) vấn đề
- Vấn đề tóm tắt không chính xác
- Không thể lặp lại đọc Vấn đề
- Phantom Reads

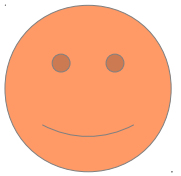
mất cập nhật

hai Processes

- Quy trình A đọc cân bằng
- Process Một trừ 10 từ _balance và sau đó viết này để DB

- Quy trình B lần đọc cân bằng
- Quy trình B trừ 100 từ _balance và sau đó viết này để DB

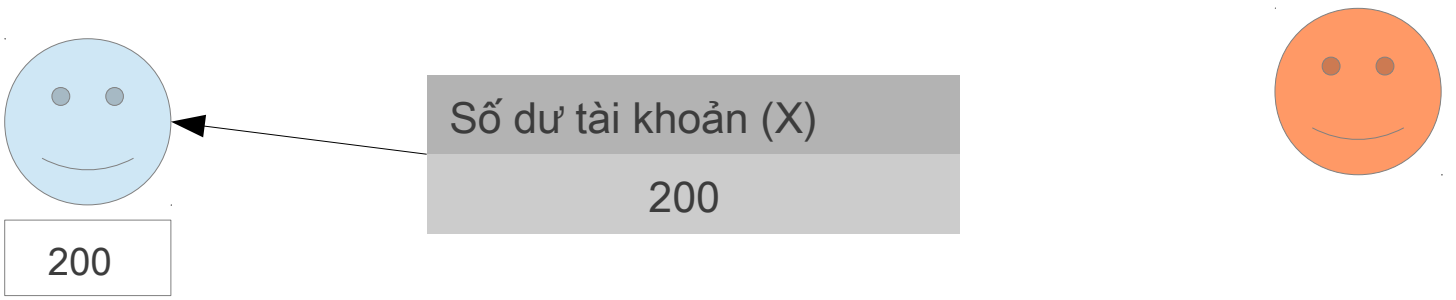
Số dư tài khoản
200



số dư cuối cùng là gì sau khi cả hai quá trình được hoàn thành?

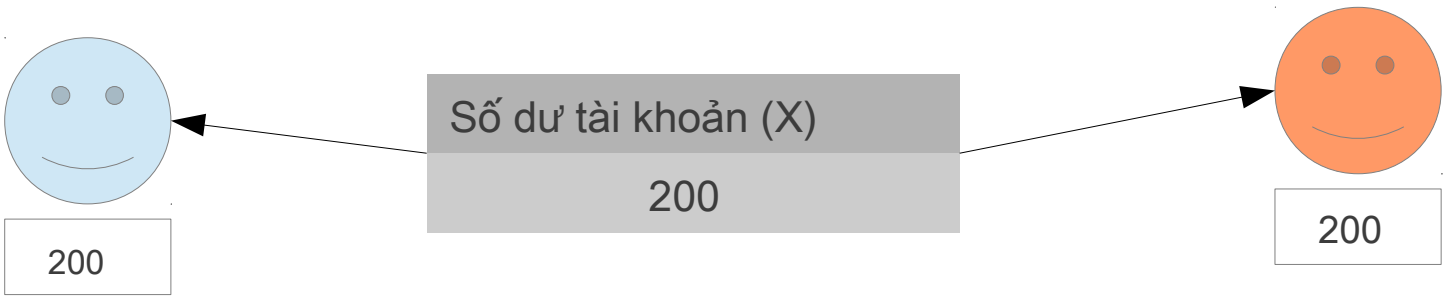
Có thể tự thực hiện

- read_item (X)



Có thể tự thực hiện

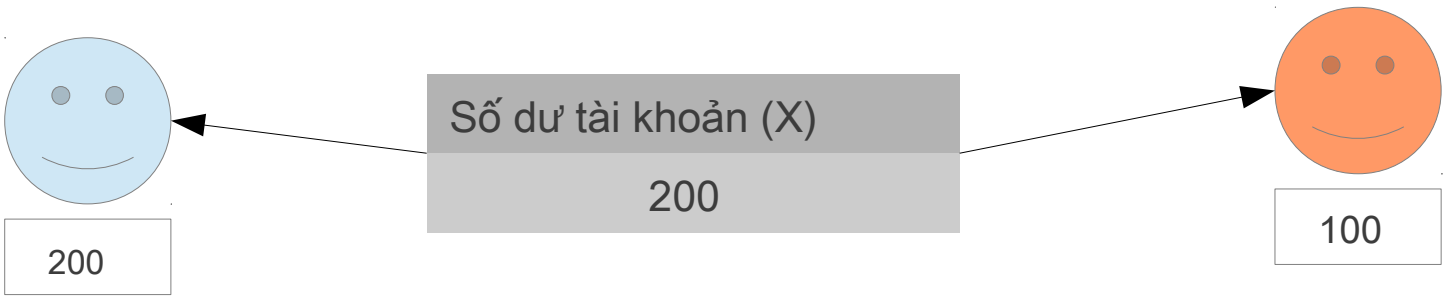
- read_item (X)
- read_item (X)



Có thể tự thực hiện

- read_item (X)

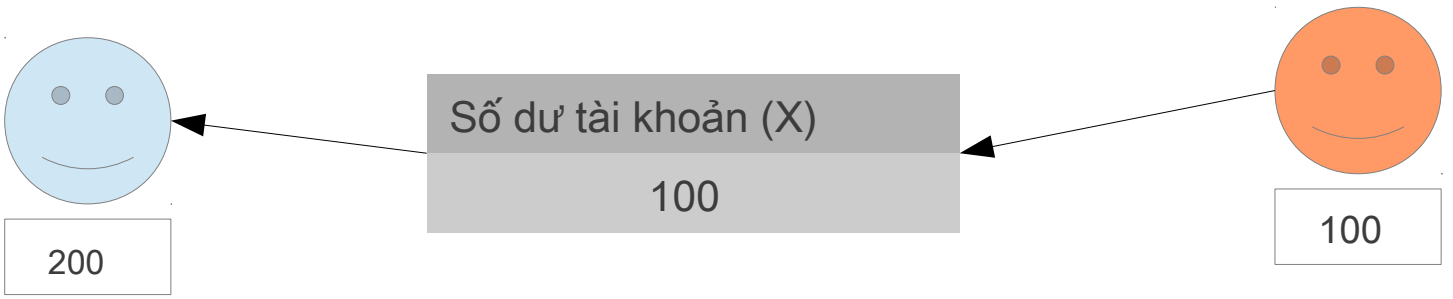
- read_item (X)
- X = X - 100



Có thể tự thực hiện

- read_item (X)

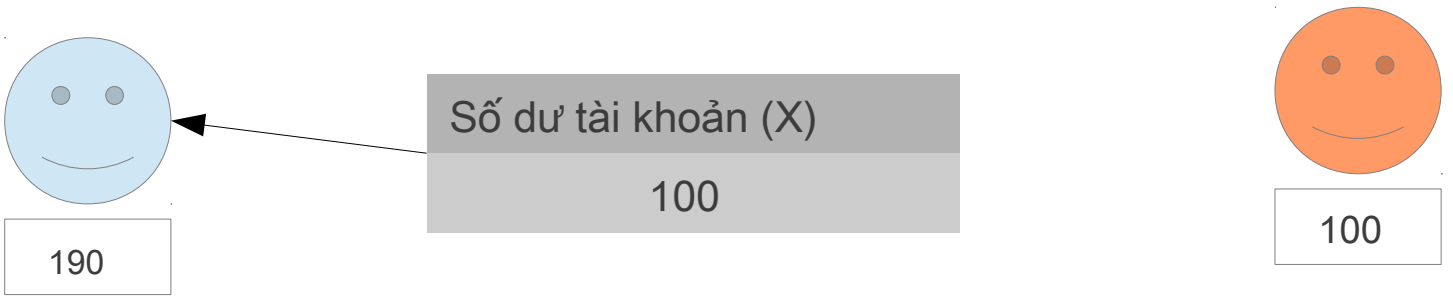
- read_item (X)
- X = X - 100
- write_item (X)



Có thể tự thực hiện

- read_item (X)
- X = X - 10

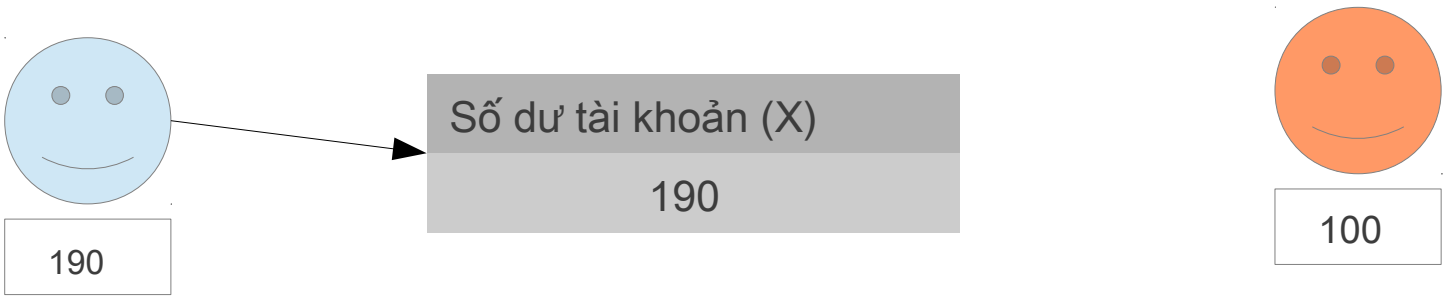
- read_item (X)
- X = X - 100
- write_item (X)



Có thể tự thực hiện

- read_item (X)
- X = X - 10
- write_item (X)

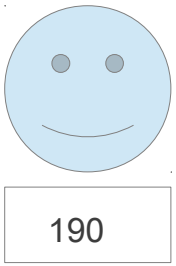
- read_item (X)
- X = X - 100
- write_item (X)



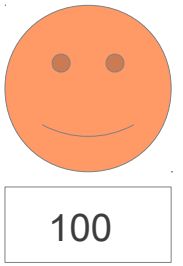
Có thể tự thực hiện

- read_item (X)
- X = X - 10
- write_item (X)

- read_item (X)
- X = X - 100
- write_item (X)



Số dư tài khoản (X)
190

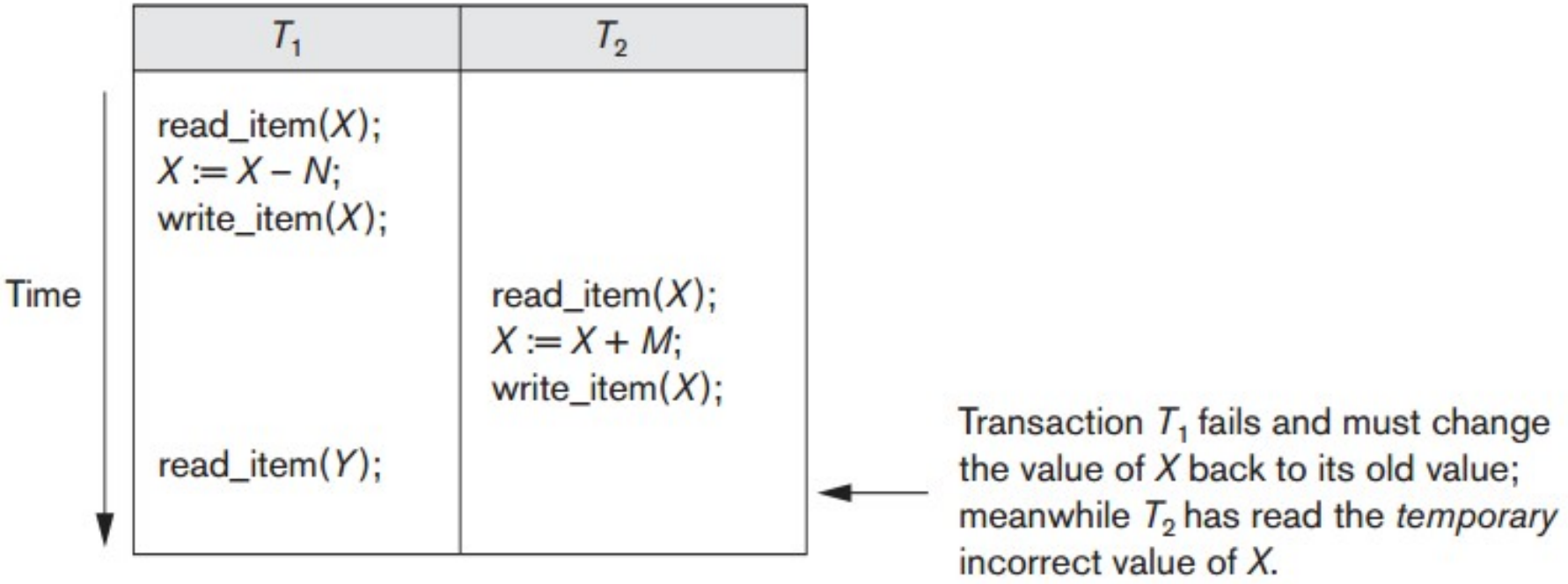


Nhưng chúng ta biết điều này là không đúng. Gì đã xảy ra với bản cập nhật của B?

Bẩn đọc sai Tóm tắt Không thể lặp lại đọc Phantom Reads

bẩn đã đọc

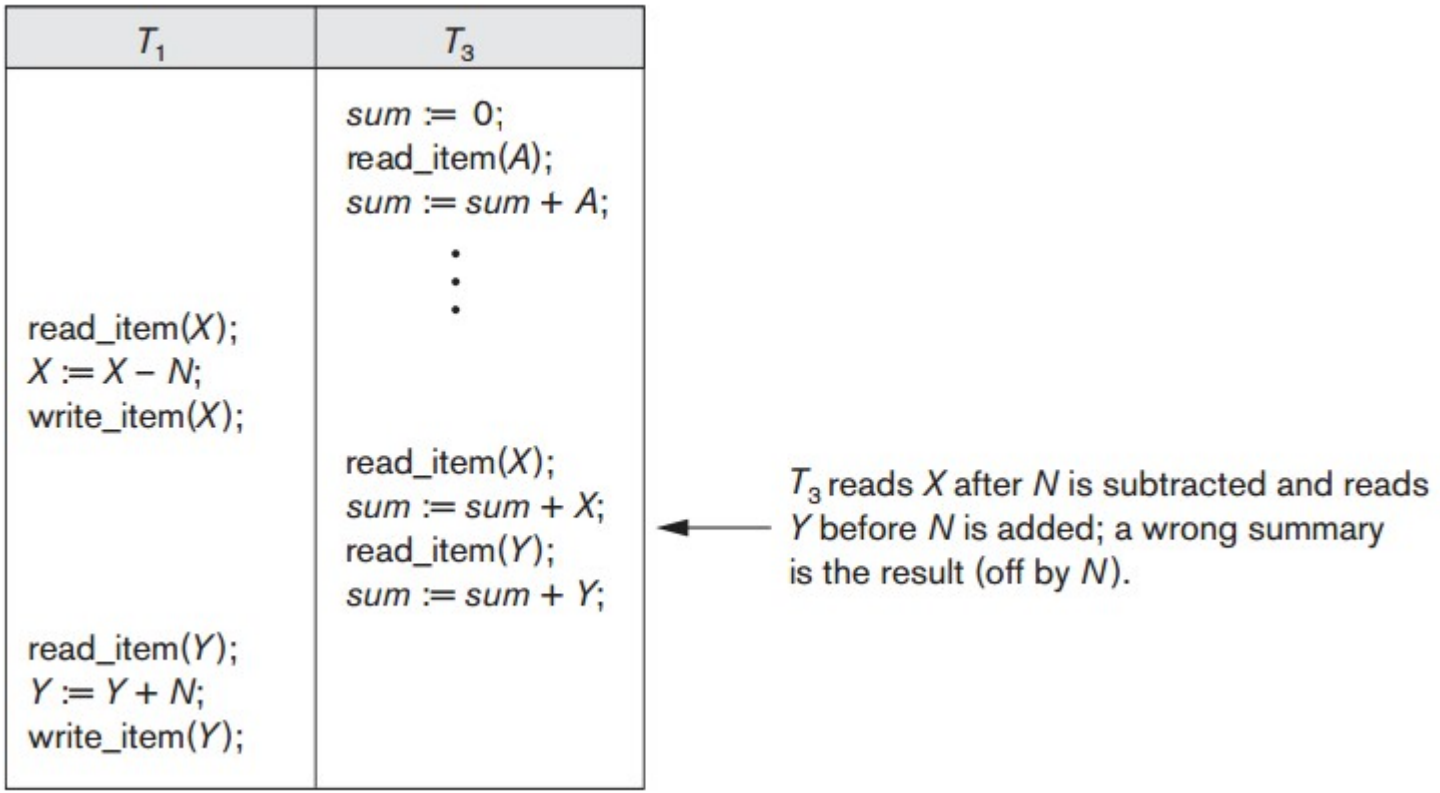
- Xảy ra khi một quá trình thất bại và một sử dụng một giá trị chưa được cam kết với cơ sở dữ liệu



- Trong ví dụ T_1 thất bại (hoặc là cuộn lại) sau khi read_item (Y) và do đó cơ sở dữ liệu nên không có dấu hiệu của T_1 đã xảy ra

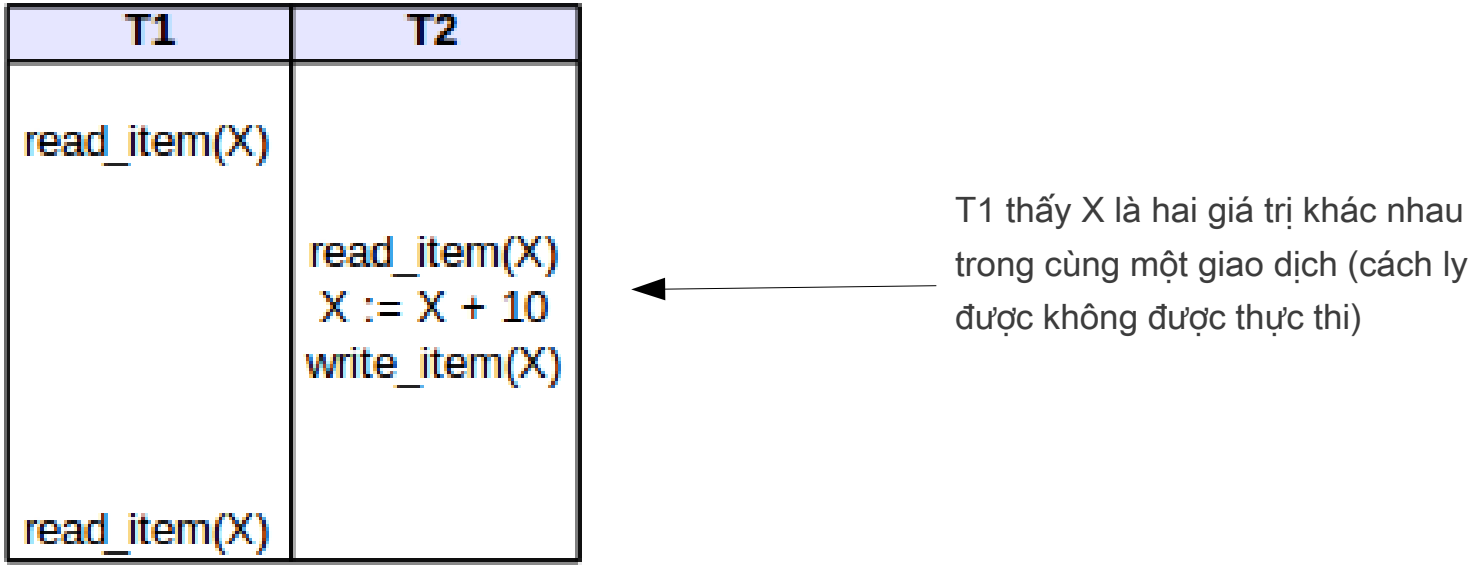
Tóm tắt thông tin không chính xác

- Xảy ra khi một quá trình được tính toán một bản tóm tắt và khác đang thay đổi các giá trị được sử dụng trong phần tóm tắt



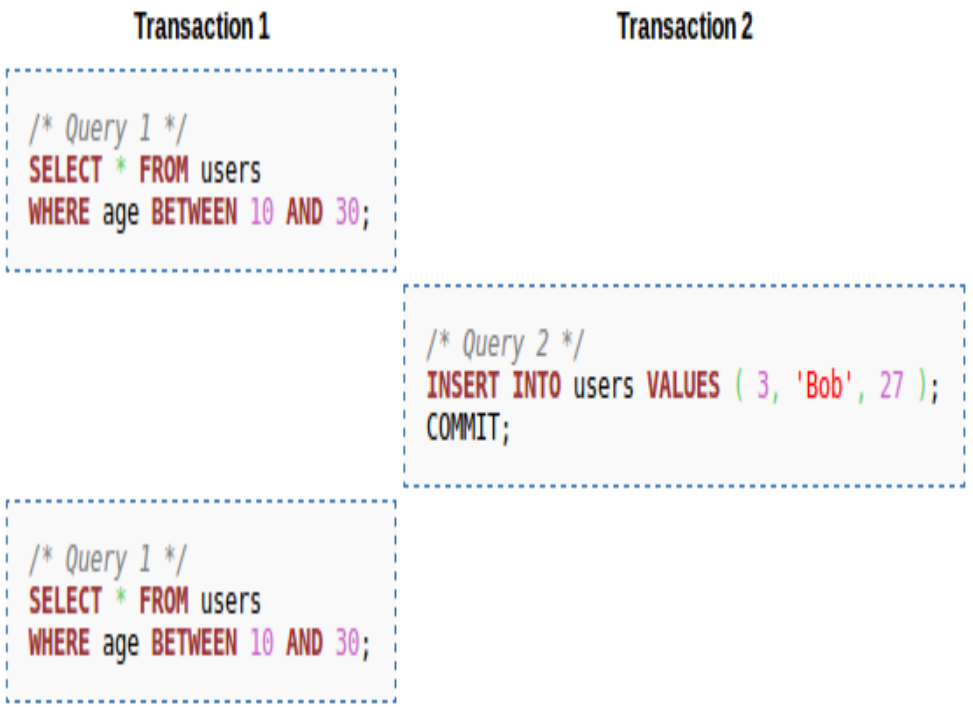
Không thể lặp lại đọc

- nơi một giao dịch T đọc cùng một mục hai lần và mục được thay đổi bởi một giao dịch T giữa hai lần đọc



Phantom đã đọc

- xảy ra khi một truy vấn phạm vi được sử dụng và một tuple chèn tuân thủ các tiêu chí truy vấn



T1 đọc sử dụng một “ở đâu” khoản. Một kỷ lục mà tuân thủ các “ở đâu” khoản được chèn và do đó xuất hiện trong đọc thứ hai.

Tóm lược

- Bạn đã từng học..
  - cách đồng thời có thể dẫn đến các vấn đề trong RDBMS
  - những gì các tính chất ACID của giao dịch là
  - rằng các giao dịch giải quyết nhiều vấn đề mà đồng thời mang lại về