Trường Đại học Khoa học Tự nhiên

Khoa Công nghệ Thông tin

Bộ môn Hệ thống Thông tin

Quản trị Cơ sở Dữ liệu

Chương VI :

GIAO TÁC

GV : TUẤN NGUYÊN HOÀI ĐỨC

E-mail : tnhduc@fit.hcmus.edu.vn

Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Giao tác (transaction)

- Khái niệm
  - ***Chúng ta xây dựng CSDL là để lưu trữ thông tin và khai thác thông tin. Công việc này gọi chung là xử lý thông tin.***
  - ***Thực tế tồn tại những bước xử lý thông tin tạo nên một đơn nguyên (atom) : những bước xử lý này hoặc là được thực hiện hết, hoặc là không thực hiện một bước nào.***
  - ***Những đơn nguyên như vậy gọi là giao tác (transaction)***

Giao tác (transaction)

- Ví dụ : chuyển khoản 100$ từ tài khoản A sang tài khoản B (50$) và tài khoản C (50$). Các bước thực hiện gồm :
  - ***Trừ 100$ khỏi tài khoản A***
  - ***Nếu số dư trong A < 0 thì khôi phục số dư cũ và ngưng***
  - ***Nếu số dư trong A >=0 thì***
    - Cộng 50$ vào tài khoản B
    - Cộng 50$ vào tài khoản C
  - ***Nếu số dư trong A = 0 thì tất toán tài khoản A***

Giao tác (transaction)

- Ví dụ :
  - ***Giả sử vừa trừ tiền khỏi tài khoản A thì sự cố kỹ thuật sảy ra và các bước tiếp theo không được thực hiện → mất 100$ → Không chấp nhận được***
  - ***Các bước xử lý nêu trên nếu đã làm thì phải làm cho hết, ngược lại thì không làm bước nào cả → chúng tạo thành một transaction, nói cách khác là một đơn vị công việc nguyên tố.***

Giao tác (transaction)

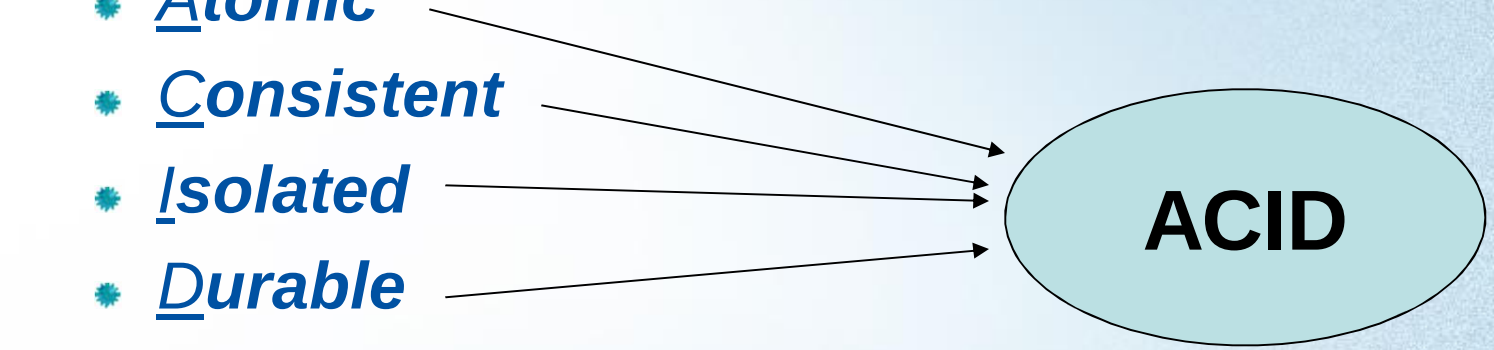
- Các tính chất :
  - ***Tính nguyên tố (Atomic) : Một giao tác là một đơn nguyên (atom), nghĩa là các công việc trong ấy không tách rời nhau được***
  - ***Tính nhất quán (Consistent) : Một giao tác có thể làm thay đổi tình trạng CSDL, nhưng không được làm mất tính nhất quán vốn có của CSDL***

Giao tác (transaction)

- Các tính chất :
  - ***Tính cô lập (Isolated) : Công việc của một giao tác không thể bị chi phối hoặc phá hoại bởi các giao tác khác***
  - ***Tính vững bền (Durable) : Kết quả làm việc của một giao tác phải được lưu trữ bền vững vào CSDL.***

Giao tác (transaction)

- Các tính chất :
  - ***Atomic***
  - ***Consistent***
  - ***Isolated***
  - ***Durable***



Giao tác (transaction)

- Các sự kiện của một giao tác
  - ***Begin tran : Giao tác bắt đầu***
  - ***Commit tran : Giao tác hoàn tất thành công***
  - ***Rollback tran : Giao tác thất bại và bị chấm dứt, mọi thay đổi nó thực hiện trên dữ liệu bị hủy bỏ, dữ liệu khôi phục trạng thái cũ. Không có tác dụng trên biến cục bộ***
  - ***Save tran (không chính quy) : Ghi nhận bền vững một phần kết quả của giao tác tính đến thời điểm save tran***

Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Xử lý đồng thời (Concurrency)

- Mô hình ứng dụng hiện nay là mô hình đa người dùng : Một CSDL lưu tại Server và nhiều clients đồng thời truy cập và thao tác trên cùng CSDL ấy
- Một client X có thể thực hiện lần lượt các giao tác Ti của nó, trong khi một client Y có thể cũng đang lần lượt thực hiện các giao tác Tj của nó.

Xử lý đồng thời (Concurrency)

- Các Ti và Tj đồng thời truy xuất và thay đổi CSDL trên server và có thể xung đột, tranh chấp lẫn nhau.
- Việc dàn xếp các xung đột, tranh chấp này sao cho ổn thỏa và bào đảm nhất quán dữ liệu gọi là xử lý đồng thời (concurrency)

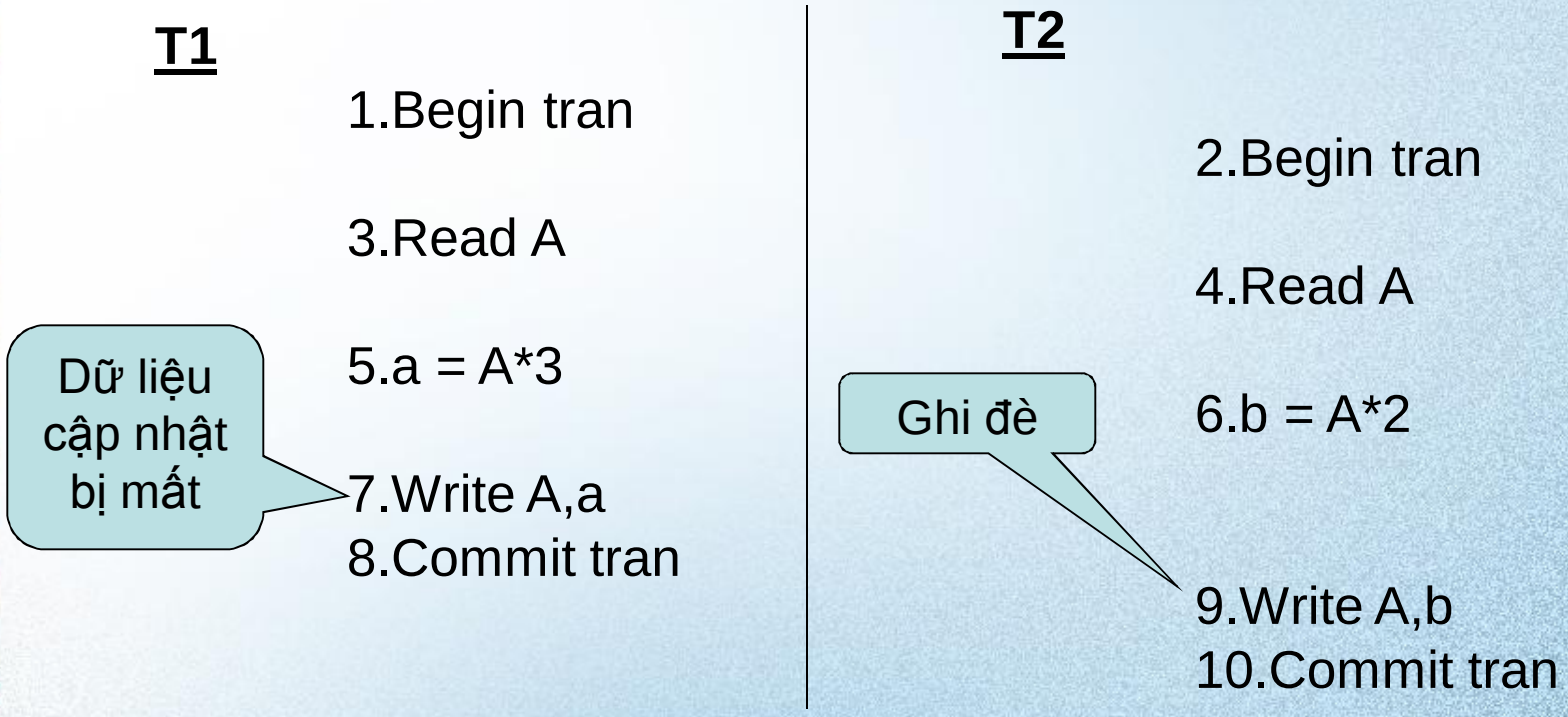
Xử lý đồng thời (Concurrency)

- Bản chất của các giao tác đồng thời : Các thao tác của những giao tác được DBMS đáp ứng theo cách mà CPU đáp ứng các lệnh từ các ứng dụng chạy trên một hệ điều hành đa nhiệm.
- Các vấn đề xử lý đồng thời : Các giao tác đồng thời sẽ được xem là ổn nếu kết quả của chúng giống như trường hợp làm tuần tự từng giao tác một.

Xử lý đồng thời (Concurrency)

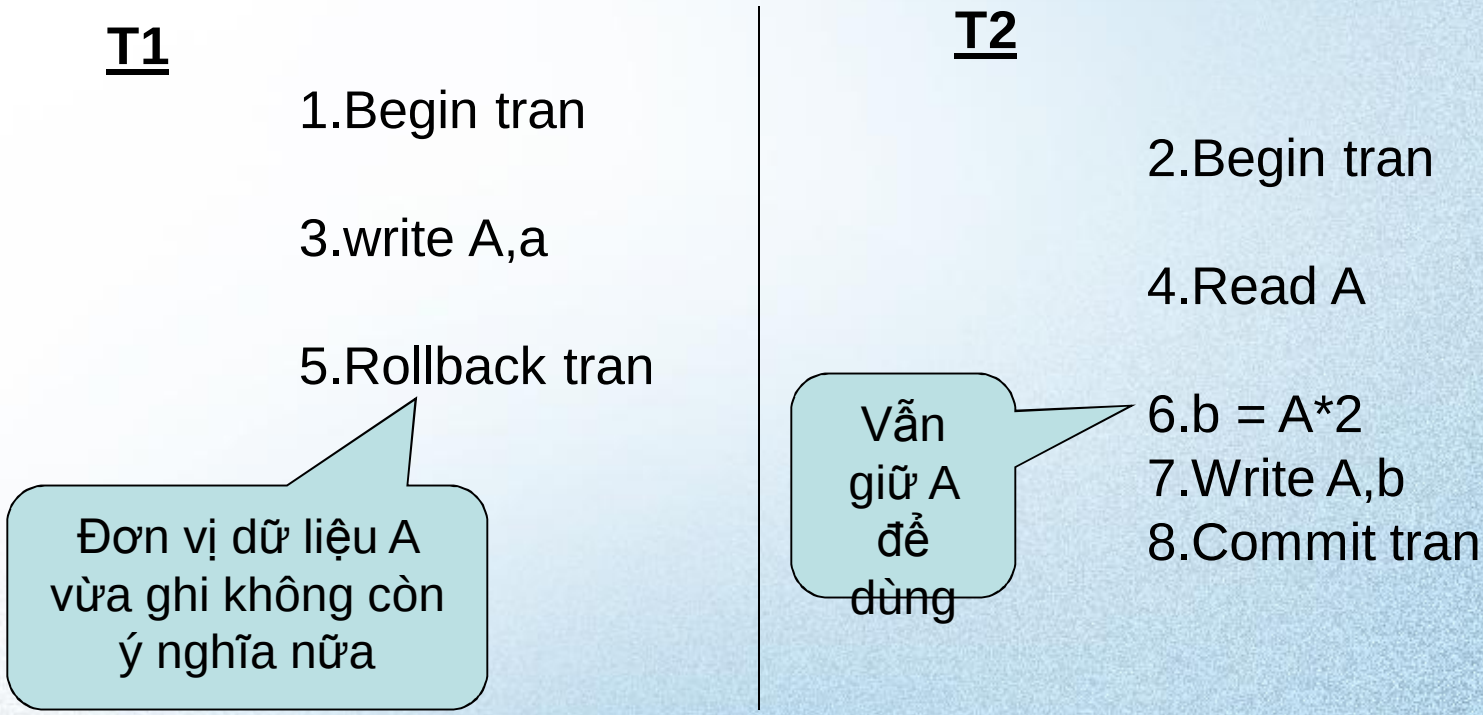
Các vấn đề xử lý đồng thời

Mất dữ liệu cập nhật (Lost update)



Xử lý đồng thời (Concurrency)

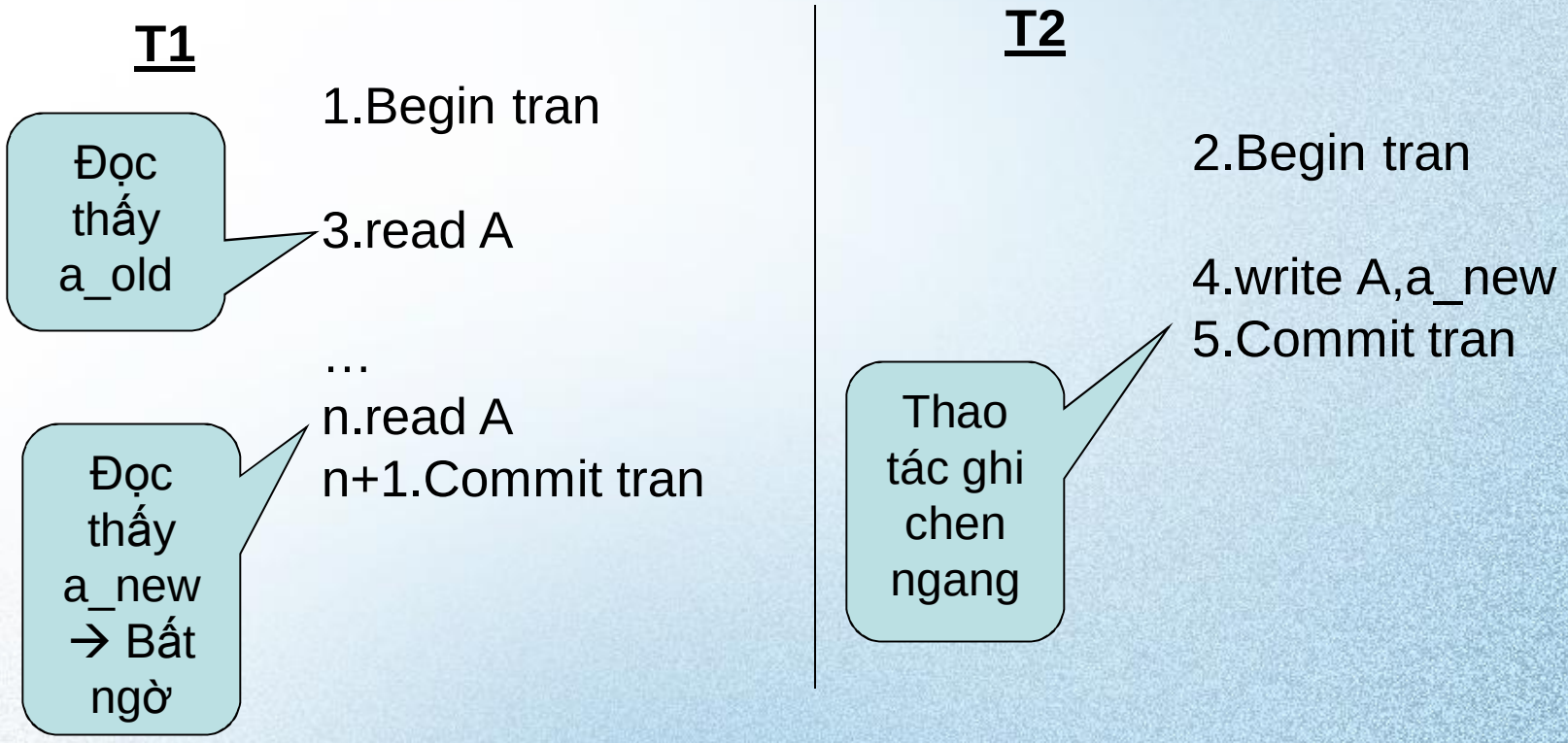
- Các vấn đề xử lý đồng thời
  - ***Đọc dữ liệu rác (Dirty read)***



Xử lý đồng thời (Concurrency)

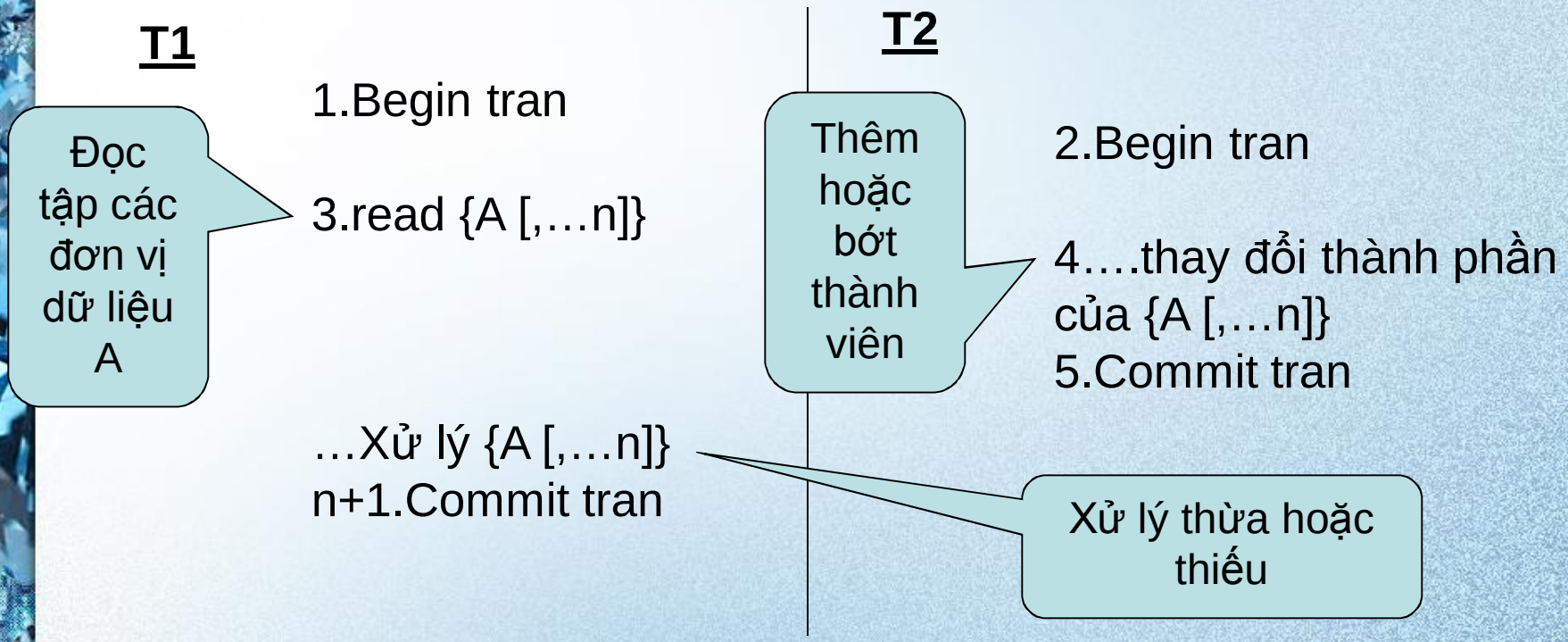
Các vấn đề xử lý đồng thời

Không thể đọc lại (Unrepeatable read)



Xử lý đồng thời (Concurrency)

- Các vấn đề xử lý đồng thời
  - ***Bóng ma (Phantom)***



Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Chế độ khóa

- Mục đích : Tránh các vấn đề về XL đồng thời vừa nêu
- Cách thức : Một giao tác T, khi thao tác trên đơn vị dữ liệu A, có thể quy định mức độ quyền hạn của các giao tác T' khác trên A bằng cách phát khóa trên A.

Chế độ khóa

- Các loại khóa
  - ***Khóa chia sẻ (shared lock) : Còn gọi là khóa đọc (read lock) . Gọi tắt : Khóa S***
  - ***Khóa dự định ghi (Intend to write lock) : Còn gọi là khóa cập nhật (update lock) . Gọi tắt : Khóa U***
  - ***Khóa độc quyền (exclusive lock) : Còn gọi là khóa ghi (write lock). Gọi tắt : Khóa X. Khóa X luôn được phát ra khi ghi, bất kể thông số hệ thống đang thiết lập thế nào.***

Chế độ khóa

- Bảng tương thích giữa các chế độ khóa

	S	U	X
S	✓	✓	✗
U	✓	✗	✗
X	✗	✗	✗

Chế độ khóa

- Khi một transaction Ti cần truy cập và thao tác trên một đơn vị dữ liệu X, nó sẽ đòi phát khóa A trên X. Nhưng khi ấy nếu transaction Tj đang giữ khóa B trên X (và khóa A với khóa B không tương thích) thì Ti phải đợi Tj giải phóng khóa B trên X nó mới phát được khóa A trên X → Hiện tượng chờ đợi lẫn nhau.

Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Khai báo tường minh giao tác

- Trong SQL Server, ta có thể khai báo tường minh các giao tác, có thể là trên một khối lệnh độc lập hay trong thân một thủ tục thường trú.
- Ngoài ra SQL Server còn có thể phát sinh các giao tác ngầm định (Ví dụ : Trigger)

Khai báo tường minh giao tác

- Các chỉ thị :
  - ***Begin tran : Đặt trước dòng lệnh đầu tiên của giao tác (1 chỉ thị duy nhất cho 1 giao tác)***
  - ***Commit tran : Đặt sau dòng lệnh cuối cùng hoàn tất giao tác (1 chỉ thị duy nhất cho 1 giao tác)***
  - ***Rollback tran : Đặt tại các vị trí kiểm lỗi hay các nhánh rẽ logic ứng với trường hợp nghiệp vụ thất bại. (nhiều chỉ thị cho 1 giao tác)***

Khai báo tường minh giao tác

- Kiểm lỗi trong giao tác
  - ***Lỗi có thể sảy ra sau các thao tác :***
    - Insert, Update (trùng khóa chính, sai kiểu dữ liệu, sai định dạng ngày tháng,…)
    - Delete (ràng buộc tồn tại,…)
    - Select (login không có quyền trên object…)
  - ***Sau mỗi thao tác trên phải kiểm lỗi bằng biến hệ thống*** *@@error* ***(=0 → không có lỗi,≠0 → có lỗi)***
  - ***Có thể không kiểm tra lệnh select để tránh làm cho giao tác quá cồng kềnh (vì lỗi này ít khi sảy ra và có thể khống chế được so với 2 loại lỗi cón lại)***

Khai báo tường minh giao tác

- Kiểm lỗi trong giao tác
  - ***Khi lỗi sảy ra (@@error ≠ 0), cần thực hiện các công việc :***
    - Báo lỗi (nếu cần) bằng các lệnh print hay raise error
    - Rollback tran (bắt buộc)
    - Nếu Tran khai báo trong SP thì thông thường phải ngưng ngay SP ấy bằng lệnh return

Khai báo tường minh giao tác

- Kiểm lỗi trong giao tác

```
If @@error <> 0
Begin
    Print '…'
    Rollback tran
    Return
End
```

Khối lệnh kiểm tra lỗi

Khai báo tường minh giao tác

- Ví dụ

```
Create proc TongTien @MaDH varchar(10)
As
    Declare @ThanhTien float
    Declare @TienThue float
    Declare @TienChietKhau float
    Declare @DonGia float,@SoLuong int
    Begin tran
    Set @SoLuong = (select SoLuong from DonHang
      where Ma = @MaDH)
    --Khối lệnh kiểm tra lỗi
```

Khai báo tường minh giao tác

- Ví dụ

```
Set @DonGia = (select DonGia from DonHang where Ma = @MaDH)
--Khối lệnh kiểm tra lỗi
Set @TienThue = (select ThueSuat from DonHang where Ma = @MaDH)
--Khối lệnh kiểm tra lỗi
Set @TienChietKhau = (select ChietKhau from DonHang where Ma = @MaDH)
--Khối lệnh kiểm tra lỗi
Set @ThanhTien = @DonGia*@SoLuong
Set @TienThue = @ThanhTien*@TienThue/100
Set @ThanhTien = @ThanhTien + @TienThue
```

Khai báo tường minh giao tác

- Ví dụ

```
Set @TienChietKhau =
  @ThanhTien*@TienChietKhau/100
Set @ThanhTien = @ThanhTien - @TienChietKhau
Update DonHang set ThanhTien = @ThanhTien
  where Ma = @MaDH
--Khối lệnh kiểm tra lỗi
Commit tran
Go
```

Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Mức cô lập cho giao tác

- Khái niệm
  - ***Mức cô lập (Isolation level) là thuộc tính cho biết một giao tác sẽ độc lập đến cỡ nào so với các giao tác khác thực hiện đồng thời với nó***
    - Mức cô lập càng thấp thì giao tác càng dễ bị chi phối, phá hoại bởi các giao tác khác, nhung nó lại xử lý nhanh do ít có hiện tượng chờ đợi sảy ra.
    - Mức cô lập càng cao thì giao tác càng khó bị chi phối, phá hoại bởi các giao tác khác, nhưng nó lại xử lý chậm do phải chờ đợi nhiều.

Mức cô lập cho giao tác

- Khái niệm
  - ***Tầm vực của Isolation level là ở mức connection chứ không phải mức transaction. Khi 1 connection N được đặt mức cô lập X thì X sẽ phát huy hiệu lực trên tất cả các transaction Ti chạy trên N.***
  - ***Phải đặt mức cô lập thích hợp cho 1 transaction trước khi bắt đầu nó (begin tran)***
  - ***Mức cô lập chỉ quyết định cách phát và giữ khóa S của transaction (vì khóa X luôn được phát ra khi ghi). Mức cô lập không quan tâm khóa U***

Mức cô lập cho giao tác

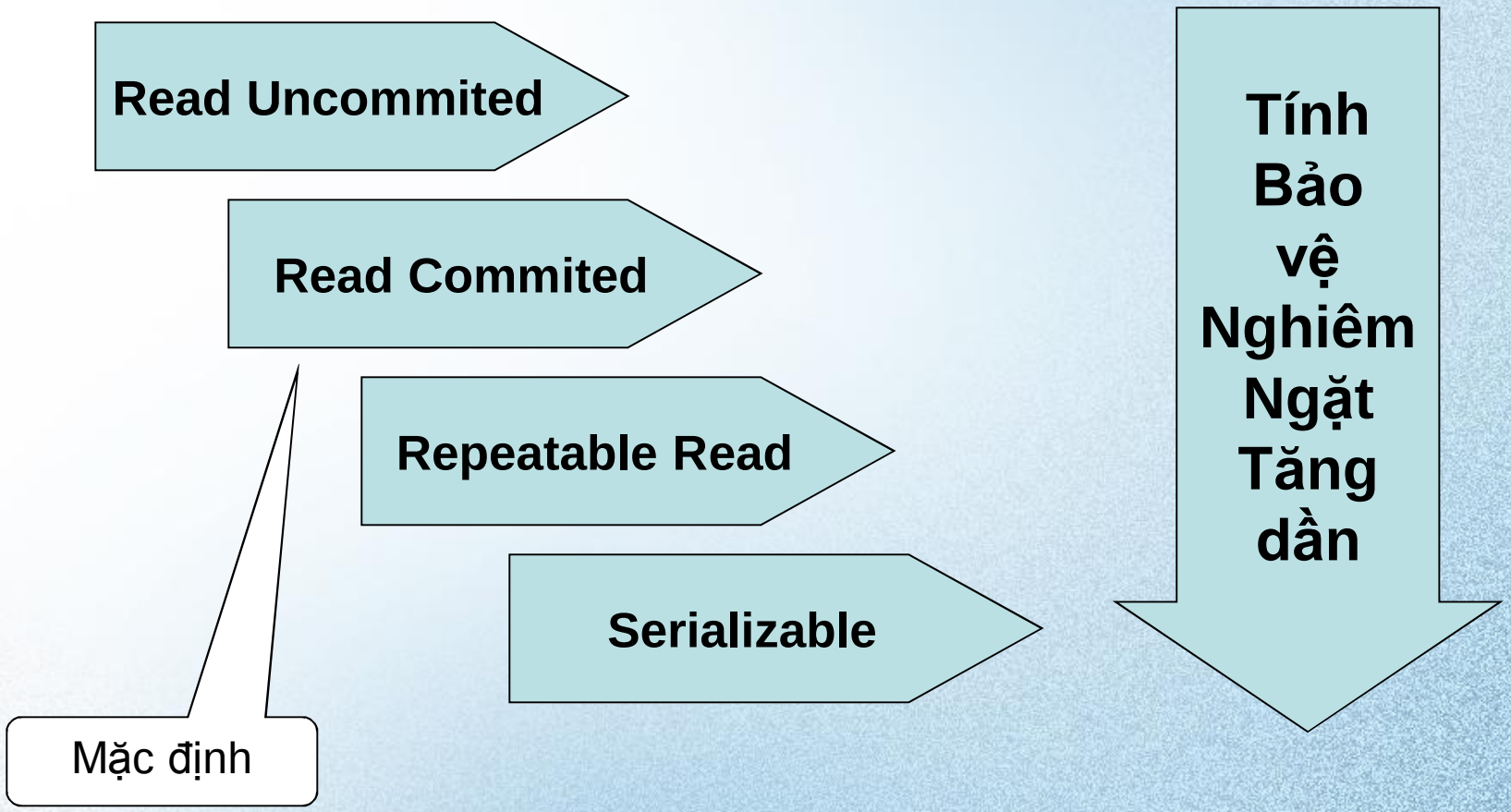
- Bốn mức cô lập mà SQL Server cung cấp
  - ***Read Uncommited : Không phát S khi đọc. Không giải quyết được bất cứ vấn đề xử lý đống thới nào.***
  - ***Read Commited : Phát S khi đọc, giải phóng S ngay sau khi đọc. Chỉ giải quyết được Dirty Read***

Mức cô lập cho giao tác

- Bốn mức cô lập mà SQL Server cung cấp
  - ***Repeatable Read : Phát S khi đọc và giữ S đến khi transaction kết thúc. Không ngăn chặn lệnh insert dữ liệu thoả điều kiện thiết lập S. Giải quyết được Dirty Read và Unrepeatable Read***
  - ***Seralizable : Giống Repeatable Read nhưng có ngăn chặn lệnh insert dữ liệu thoả điều kiện thiết lập S. Giải quyết được Dirty Read và Unrepeatable Read và Phantom***

Mức cô lập cho giao tác

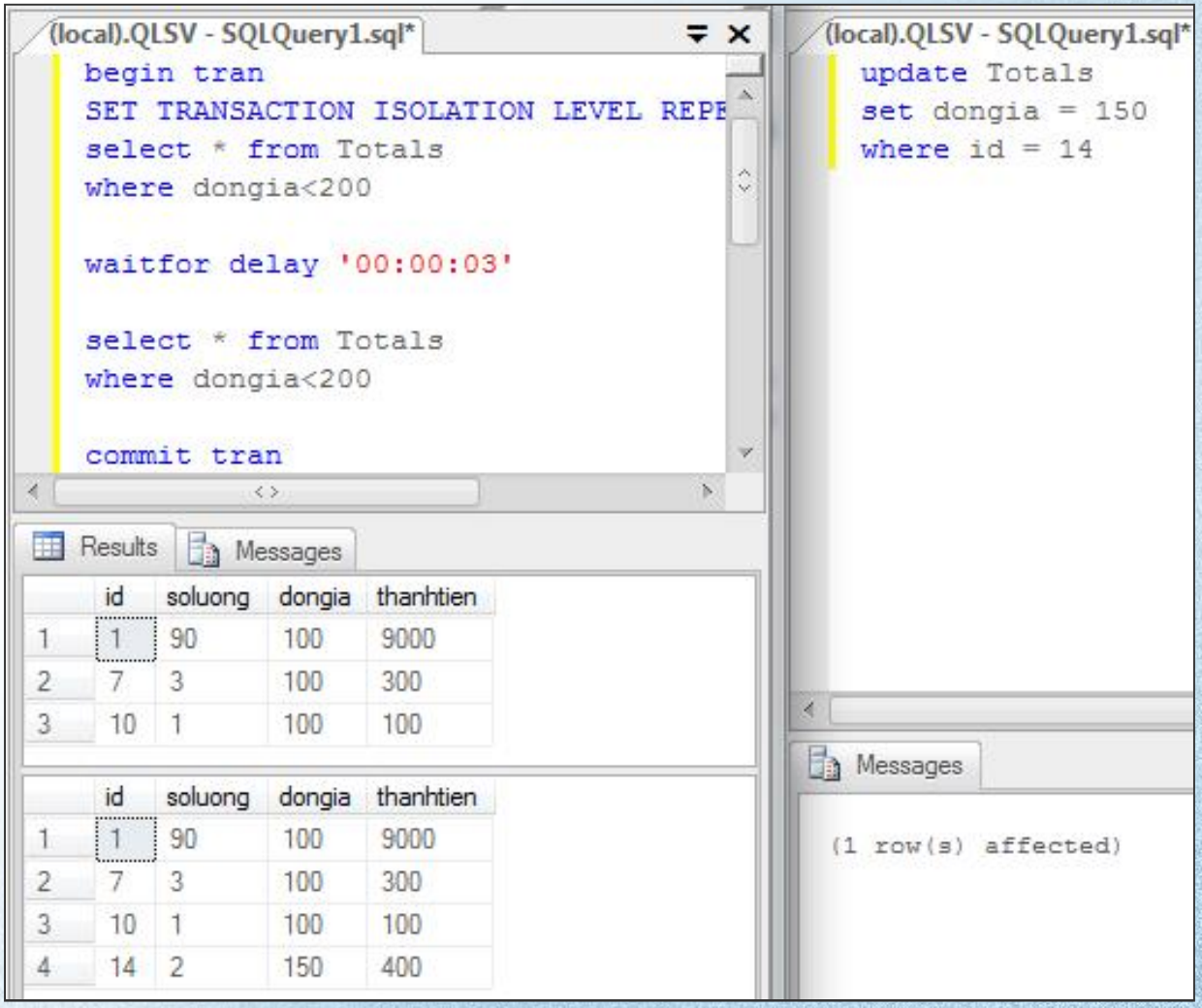
- Bốn mức cô lập mà SQL Server cung cấp



Mức cô lập cho giao tác

- Thiết lập :
  - ***Đặt lệnh thiết lập cần thiết trước khi bắt đầu giao tác. Không nên ỉ lại vào thiết lập mặc định***
  - ***Lệnh thiết lập :***
    - **Set transaction Isolation level** Tên_MứcCôLập
  - ***Ví dụ :***
    - **Set transaction Isolation level** Read Commited
    - **Set transaction Isolation level** Serializable

Mức cô lập cho giao tác



Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Các cấp độ khóa

- Đặt vấn đề : Mức cô lập là chưa đủ
  - ***Mức cô lập quyết định cách phát và giữ khóa S trong một transaction và có hiệu lực trên tất cả các thao tác đọc trong transaction đó.***
  - ***Thực tế, ta cần phát và giữ khóa S theo các cách khác nhau cho các thao tác đọc khác nhau trong cùng một transaction***
  - ***Ngoài ra, ta cũng cần dùng nhiều dạng khóa linh động hơn là chỉ khóa S đơn giản***

Các cấp độ khóa

- Khái niệm
  - ***Cấp độ khóa là các loại khóa khác nhau (không chỉ khóa S) được gắn vào từng table trong mệnh đề from của từng thao tác select***
  - ***Ngoài lệnh select, cấp độ khóa còn có thể gắn vào các câu lệnh cập nhật, tuy nhiên ở đây ta chỉ quan tâm câu select***

Các cấp độ khóa

- Các cấp độ khóa
  - ***Read Uncommited / No lock***
  - ***Read Commited (mặc định)***
  - ***Repeatable***
  - ***Serializable / Hold lock***
  - ***Updlock***
  - ***Tablock***
  - ***TablockX***
  - ***ReadPast……***

Các cấp độ khóa

- Cách thiết lập

  Select …

  From {Tab1 Alias1** with **Lock_mode [,…n]} [,…n]

  Where …

- Ví dụ :

  *Select **SV.HoVaTen, K.TenKhoa***

  *From **SinhVien SV** with **ReadCommited**,*

  Khoa K** with **Updlock

  *Where **SV.Khoa = K.Ma** And **Year(SV.NgaySinh) >= 1983***

Các cấp độ khóa

- Phối hợp với Isolation Level
  - ***Trong transaction luôn có các thao tác yêu cầu bảo vệ nghiêm ngặt và các thao tác ít yêu cầu bảo vệ nghiêm ngặt***
  - ***Dùng Isolation level ứng với yêu cầu bảo vệ ít nghiêm ngặt nhất***
  - ***Bổ sung lock mode vào các thao tác yêu cầu bảo vệ nghiêm ngặt hơn mức mà Isolation level đó cung cấp.***

Các cấp độ khóa

- Khóa với dữ liệu trong cursor
  - ***Nếu cursor là loại tĩnh (static) thì các đơn vị dữ liệu đọc ra sẽ được lock ngay khi vừa Open cursor***
  - ***Nếu cursor là loại động (Dynamic) thì fetch đến đâu sẽ khóa đến đó***
  - ***Cách phát khóa và giữ khóa là do mức cô lập của connection và các lock mode trong câu select định nghĩa cursor quyết định***

Nội dung trình bày

- Giao tác (Transaction)
- Xử lý đồng thời (Concurrency)
- Chế độ khóa
- Khai báo tường minh giao tác
- Mức cô lập
- Các cấp độ khóa
- Dead-lock

Dead lock

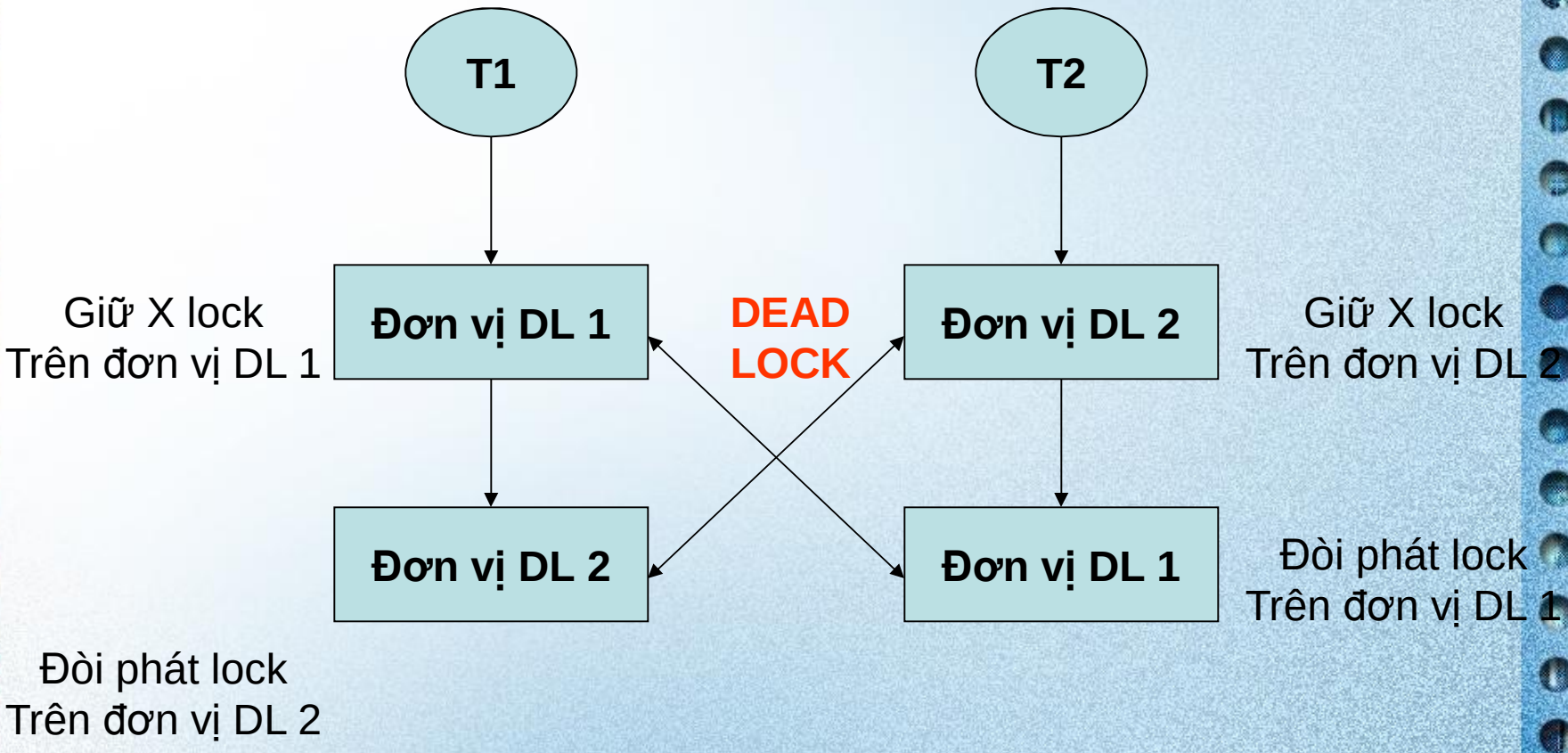
- Khái niệm
  - ***Khi xử lý đồng thời, không tránh khỏi việc transaction này phải chờ đợi transaction khác***
  - ***Nếu vì lý do gì đó mà hai transaction lại chờ lẫn nhau vĩnh viễn, không cái nào trong hai có thể hoàn thành được thì ta gọi đó là hiện tượng*** *Dead Lock*

Dead lock

- Phân loại
  - ***Cyclic Deadlock***
  - ***Conversion Deadlock***

Dead lock

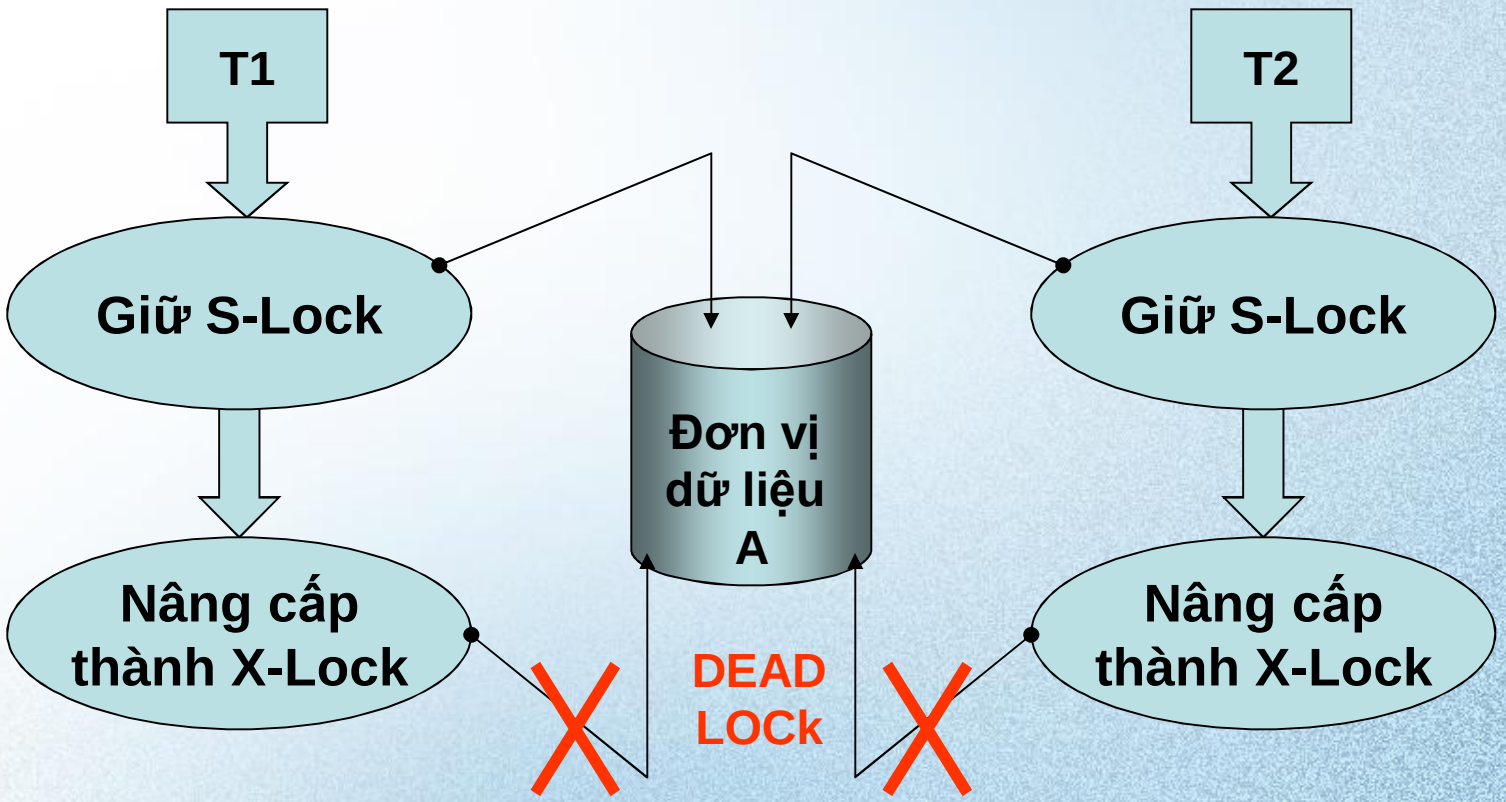
- Phân loại
  - ***Cyclic Deadlock***



Dead lock

Phân loại

Conversion Deadlock



Dead lock

- Khi dead lock sảy ra
  - ***SQL Server sẽ chọn 1 trong 2 transaction gây dead lock để hủy bỏ, khi đó transaction còn lại sẽ được tiếp tục thực hiện cho đến khi hoàn tất***
  - ***Transaction bị chọn hùy bỏ là transaction mà SQL ước tính chi phí cho phần việc đã làm được ít hơn transaction còn lại.***

Hết chương VI