

# TRANH CHẤP ĐỒNG THỜI

Trình bày : Hồ Thị Hoàng Vy

1

## Nội dung trình bày

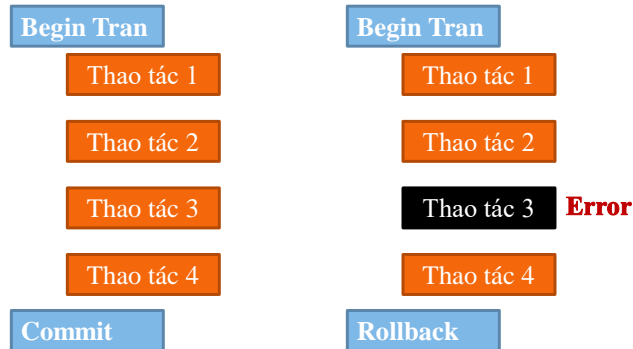
- ✧ Giao tác
- ✧ Tranh chấp đồng thời
- ✧ Xử lý tranh chấp đồng thời
- ✧ Deadlock
- ✧ Xử lý deadlock

2

2

## Giao Tác (Transaction)

☞ Giao tác là tập các thao tác có thứ tự, truy xuất dữ liệu trên CSDL. Giao tác chuyển CSDL từ trạng thái nhất quán này sang trạng thái nhất quán khác



3

3

## Giao Tác

☞ **Tính chất của giao tác**

- Tính nguyên tử - Atomicity
- Tính nhất quán – Consistency
- Tính cô lập – Isolation
- Tính bền vững – Durability

4

4

# Tính chất giao tác

## Tính nguyên tử - Atomicity

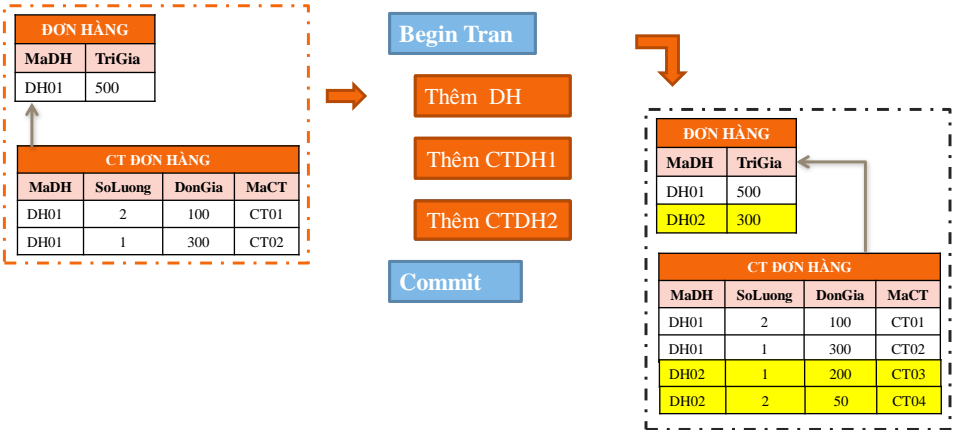


5

5

# Tính chất giao tác

## Tính nhất quán - Consistency

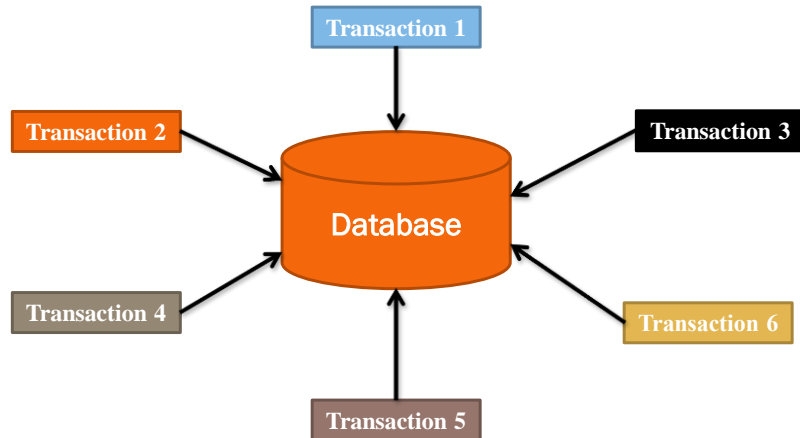


6

6

# Tính chất giao tác

## ☞ Tính cô lập - Isolation

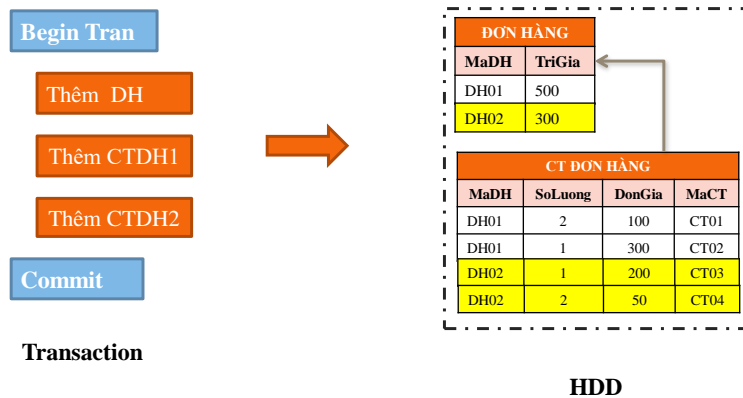


7

7

# Tính chất giao tác

## ☞ Tính bền vững – Durability



8

8

# Giao tác (Transaction)

## Cú pháp khai báo

### Đơn giản

```
BEGIN TRANSACTION

-- STATEMENTS GO HERE

COMMIT
```

### Có xử lý lỗi

```
BEGIN TRANSACTION

BEGIN TRY
    -- STATEMENTS GO HERE
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION
END CATCH

COMMIT
```

```
BEGIN TRANSACTION

-- STATEMENTS GO HERE
IF @@ERROR <> 0
    ROLLBACK TRANSACTION

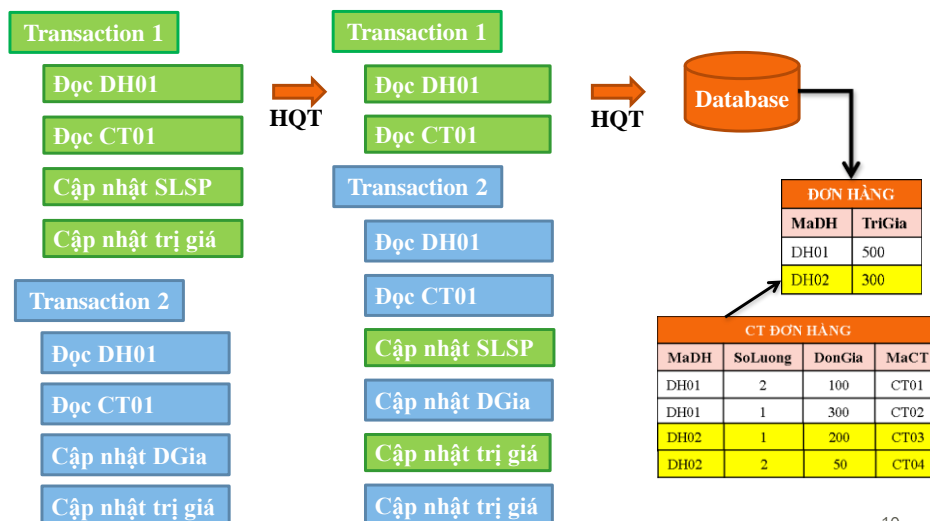
COMMIT
```

@@trancount – biến quản lý số giao tác đang chạy

9

9

# Tranh chấp đồng thời



10

10

## Tranh chấp đồng thời

Connection 1- Transaction 1	Connection 2 – Transaction 2	Nhận xét
Đọc	Đọc	Không tranh chấp
Đọc	Ghi	Xảy ra tranh chấp
Ghi	Đọc	Xảy ra tranh chấp
Ghi	Ghi	Tại một thời điểm chỉ có 1 thao tác ghi trên một đơn vị dữ liệu

⇒ Cho 2 giao tác, nếu có ít nhất một thao tác ghi trên cùng đơn vị dữ liệu sẽ gây ra tranh chấp

⇒ Tranh chấp có thể dẫn đến sai sót trên CSDL

12

12

## Tranh chấp đồng thời

### ⌘ Vấn đề tranh chấp đồng thời

- Lost Update
- Dirty Read
- Unrepeatable Read
- Phantom Read

13

13

Lost update		TAIKHOAN	
		MaTK	Sodu
		TK01	300
		TK02	200

**YC01:**

- T<sub>1</sub> Rút tiền từ tài khoản
- T<sub>2</sub> Gửi tiền vào tài khoản.

T <sub>1</sub>	T <sub>2</sub>
Set @sodu = (Select sodu From TAIKHOAN Where MaTk = 'TK01')	
100	100
50	Set @sodu = (Select sodu From TAIKHOAN Where MaTk = 'TK01')
Set @sodu = @sodu - @sotienrut	
Update TAIKHOAN Set SoDu = @sodu Where MaTK = 'TK01'	300
50	Set @sodu = @sodu + @sotiengui
	Update TAIKHOAN Set SoDu = @sodu Where MaTK = 'TK01'
	300
Commit	
	Commit

Mất dữ liệu cập nhật

14

Dirty read		TAIKHOAN	
		MaTK	Sodu
		TK01	200
		TK02	200

**YC02:**

- T<sub>1</sub> Cập nhật thông tin số dư tài khoản
- T<sub>2</sub> Vắn tin tài khoản.

T <sub>1</sub>	T <sub>2</sub>
Set @sodu = (Select sodu From TAIKHOAN Where MaTK = 'TK01')	
100	Độc dữ liệu rác
Set @sodu = @sodu + @sotien	
Update TAIKHOAN Set SoDu = @sodu Where MaTK = 'TK01'	
200	
	Select * From TAIKHOAN WITH(NOLOCK) Where MaTK = 'TK01'
	200
Rollback	Commit
Hủy	

15

15

# Unrepeatable Read

**YC03:**

- T<sub>1</sub> Đăng nhập
- T<sub>2</sub> Đổi mật khẩu

KHACHHANG	
MAKH	MATKAU
KH01	222
KH02	123
KH03	AAA

T <sub>1</sub>	T <sub>2</sub>
If (Not Exists (Select (MaKH = 'KH01')) Print N' Sai tên đăng nhập' return	Ok
If (Not Exists (Select (MaKH = 'KH01', MatKau = '1112')) Print N' Sai mật khẩu' return	Ok
	Update KHACHHANG('KH01', '222')
Select (MaKH = 'KH01', MatKau = '1112') Print N' Đăng nhập thành công'	Không xuất được dòng nào
Commit	Commit

**Không thể  
đọc lại dữ liệu**

16

# Phantom Read

**YC03:**

- T<sub>1</sub> Khuyến mãi các SP tồn > 300
- T<sub>2</sub> Thêm sản phẩm mới

SANPHAM	
MASP	SLTON
SP01	500
SP02	400
SP03	200
SP06	500

T <sub>1</sub>	T <sub>2</sub>
SET @SLSP = SANPHAM::count (MaSP, SLTON > 300) Print N'Tổng SP:' + @SLSP	2
Duyệt Dynamic Cur (SP01, SP02) Cur 01 : Print SP01	Bóng ma
	Insert SANPHAM ('SP06', 500)
Cur02 : Print SP02	
Cur03 : Print SP06	
Commit	Commit

18

18



## Giải pháp xử lý tranh chấp đồng thời

- ✧ **Kỹ thuật khóa** : các giao tác muốn đọc/ghi trên các đơn vị dữ liệu phải phát ra yêu cầu xin khóa trên đơn vị dữ liệu đó.
- ✧ **Mức cô lập** : là các thiết lập trong giao tác quy định việc xin khóa/giữ khóa của những thao tác đọc/ghi lên đơn vị dữ liệu.
- ✧ **Khóa trực tiếp trong từng câu lệnh**

19

19

## Kỹ thuật khóa

- ❖ **Không đặt khóa (*Nolock*)**
- ❖ **Khóa chia sẻ (*shared lock*)** :
  - Còn gọi là khóa đọc (*read lock*) . Gợi tắt : Khóa S
  - Khi đọc một đơn vị dữ liệu SQL tự thiết lập shared lock trên đơn vị dữ liệu đó (trừ trường hợp sử dụng *NoLock*).
  - Shared lock có thể được thiết lập trên 1 trang, 1 bảng, hay một dòng dữ liệu.
  - Nhiều giao tác có thể đồng thời giữ shared lock trên cùng 1 đơn vị dữ liệu.
  - Không thiết lập được Exclusive Lock trên đơn vị dữ liệu đang có shared Lock.
  - Shared lock thường được giải phóng ngay khi sử dụng xong dữ liệu được đọc, trừ khi thiết lập giữ shared lock đến hết giao tác.

20

20

## Kỹ thuật khóa

### ❖ Khóa cập nhật (*update lock*)

- Còn gọi là *Intend to write lock*. Gọi tắt: Khóa U.
- Là chế độ trung gian của shared lock và Exclusive lock.
- Update lock không ngăn shared lock.
- Dùng khi có dự định ghi dữ lại dữ liệu đã đọc.
- Tại một thời điểm chỉ được phát 1 Update lock trên 1 đơn vị dữ liệu => Giúp tránh Deadlock khi cần chuyển từ shared lock lên Exclusive lock trên cùng đơn vị dữ liệu.

21

21

## Kỹ thuật khóa

### ❖ Khóa độc quyền (*exclusive lock*)

- Còn gọi là khóa ghi (Write Lock). Viết tắt : X
- SQL Server tự thiết lập Exclusive lock khi thực hiện thao tác ghi (insert, update, delete) trên 1 đơn vị dữ liệu.
- Exclusive Lock luôn được giữ đến hết giao tác.
- Tại 1 thời điểm, chỉ có tối đa 1 giao tác được quyền giữ Exclusive Lock trên 1 đơn vị dữ liệu.
- Không thể thiết lập Exclusive Lock trên đơn vị dữ liệu đang có Shared Lock..

22

22

## Quy tắc đọc/ ghi trên CSDL

1. Giao tác T đọc đơn vị dữ liệu (đvdl) A  $\Rightarrow$  T xin khóa đọc trên A. Nếu hệ thống cấp khóa đọc cho T thì T được phép đọc đvdl A.
2. Giao tác T thực hiện việc ghi lên đvdl A  $\Rightarrow$  T xin khóa ghi trên A. Nếu hệ thống cấp khóa ghi cho T thì T được phép ghi lên đvdl A.
3. Tại một thời điểm, chỉ có tối đa 1 giao tác giữ khóa ghi trên 1 đvdl.
4. Tại một thời điểm, có thể có nhiều giao tác cùng giữ khóa đọc trên đvdl
5. Nếu một giao tác T đang giữ khóa ghi trên A thì đến hết giao tác (rollback/commit) thì T mới trả khóa ghi.
6. Giao tác T đang giữ khóa ghi trên A  $\Rightarrow$  Giao tác khác muốn xin khóa đọc trên A thì phải chờ.
7. Giao tác T đang giữ khóa đọc trên A  $\Rightarrow$  Giao tác khác muốn xin khóa ghi trên A thì phải chờ.

23

23

## Where are locks placed in the database?

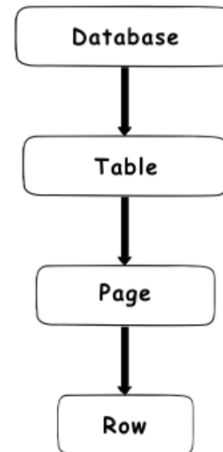
Resource	Description
RID (RowID)	It locks the single row in a table.
Table	It locks the whole table, including all data and indexes.
Key	It is used for row lock within an index such as primary key, candidate key, secondary key, etc.
Page	It is used to lock an 8-KB page in a database. If a lock is applied on a particular page, another user can't update data on it.
Extent	It locks the contiguous group of 8 data pages.
Database	It locks the whole database for users who have only read permission.
File	A lock is applied to a database file.
Application	It locks the application-specific resources.
Metadata	Lock is applied to metadata.
Object	A lock is placed on anything with a sys.all objects entry, such as a stored procedure or a view.

24

24

## Locking Hierarchy

- ☞ **Locking hierarchy** is used while reading or modifying the data.
- ☞ There are mainly two locking levels available on which lock mode is applied: **Row level** and **Table level**



25

25

## Lock Compatibility Matrix

- ☞ We can apply the following three lock modes at the row level:
  - Exclusive (X)
  - Shared (S)
  - Update (U)

26

26

## Kỹ thuật khóa

	Nolock	Shared Lock (S)	Update Lock (U)	Exclusive lock (X)
Select	☒	☑	☒	☒
Update	☒	☒	☑	☑
Insert	☒	☒	☒	☑
Delete	☒	☒	☑	☑

☑ Phát khóa tương ứng

☒ Không phát khóa tương ứng

27

27

## Bảng tương thích khóa

	Shared Lock (S)	Update Lock (U)	Exclusive lock (X)
Shared Lock (S)	☑	☑	☒
Update Lock (U)	☑	☒	☒
Exclusive lock (X)	☒	☒	☒

☑ Cho phép (tương thích)

☒ Không cho phép (không tương thích)

Chưa giải quyết  
được hết vấn đề  
tranh chấp

28

28

## lock modes at the table level

☞ We can apply the following five lock modes at the table level:

- Exclusive (X)
- Shared (S)
- Intent exclusive (IX)
- Intent shared (IS)
- Shared with intent exclusive (SIX)

29

29

## lock modes at the table level

Modes	Exclusive (X)	Shared (S)	Intent Exclusive (IX)	Intent Shared (IS)	Shared with intent exclusive (SIX)
<b>Exclusive (S)</b>	X	X	X	X	X
<b>Shared (S)</b>	X		X	✓	X
<b>Intent Exclusive (IX)</b>	X	X	✓	✓	X
<b>Intent Shared (IS)</b>	X	✓	✓	✓	✓
<b>Shared with intent exclusive (SIX)</b>	X	X	X	✓	X

30

30

## Isolation levels (mức cô lập)

### 🔗 Mức cô lập của giao tác kiểm soát:

- Lock có được sử dụng khi dữ liệu được đọc hay không và loại Lock được sử dụng.
- Read lock tồn tại bao lâu?
- Liệu 1 thao tác đọc tham chiếu đến các hàng có thể được sửa bởi 1 transaction khác hay không?

31

31

## Mức cô lập

**Read Uncommitted**

Đọc dữ liệu khi chưa commit

**Read Committed**

Chỉ đọc dữ liệu đã commit(Default)

**Repeatable Read**

Cho phép đọc lại dữ liệu

**Serializable**

Cho phép đọc tuần tự

32

32

## SET TRANSACTION ISOLATION LEVEL

☞ Controls the locking and row versioning behavior of Transact-SQL statements issued by a connection to SQL Server:

☞ SET TRANSACTION ISOLATION LEVEL

☞ {

- READ UNCOMMITTED |
- READ COMMITTED |
- REPEATABLE READ |
- SERIALIZABLE

☞ }

☞ Ex: SET TRANSACTION ISOLATION LEVEL READ  
UNCOMMITTED

33

33

## Mức cô lập

### ❖ Read Uncommitted

- Đặc điểm: Không phát S khi đọc do đó không phải chờ khi đọc dữ liệu.
- Lỗi:
  - Đọc dữ liệu rác
  - Không đọc lại được dữ liệu
  - Bóng ma
  - Mất dữ liệu đã cập nhật
- Giải quyết: Không giải quyết được bất cứ vấn đề xử lý đồng thời nào.

34

34



## Mức cô lập

<b>T<sub>1</sub></b>	<b>T<sub>2</sub></b>
Set → Read Uncommitted	
Select (TAIKHOAN) T <sub>1</sub> : Không xin khóa	
	Update TAIKHOAN('TK01', 300) T <sub>2</sub> : Xin cấp X-lock trên bảng TAIKHOAN SQL : Cấp X-lock trên bảng TAIKHOAN
Commit	Commit

35

35

## Mức cô lập

**Mức cô lập được gán trên giao tác T<sub>2</sub> (giao tác đọc dữ liệu)**

```
CREATE PROC USP_VANTINTK
    @MATK CHAR(10)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL READ UNCOMMITTED
    DECLARE @SODU FLOAT = 0
    SELECT *
    FROM TAIKHOAN
    WHERE MATK = @MATK
COMMIT TRAN
```

36

36

## Mức cô lập

### ❖ Read Committed

- Đặc điểm: Phát S khi đọc và trả khóa ngay khi đọc xong.
- Lỗi:
  - Không đọc lại được dữ liệu
  - Bóng ma
  - Mất dữ liệu cập nhật
- Giải quyết: Giải quyết được **Dirty Read**

37

37

## Mức cô lập

T <sub>1</sub>	T <sub>2</sub>
	Set → Read Committed
Update TAIKHOAN('TK01', 300) T <sub>1</sub> : Xin khóa X SQL : Cập khóa X	
	Select (TAIKHOAN) T <sub>2</sub> : Xin khóa S SQL : Không cấp khóa S do T1 đang giữ khóa X
Commit	Commit

38

38

## Mức cô lập

Mức cô lập được gán trên giao tác  $T_2$  (giao tác đọc dữ liệu)

```
CREATE PROC USP_VANTINTK
    @MATK CHAR(10)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL READ COMMITTED
    DECLARE @SODU FLOAT = 0
    SELECT *
    FROM TAIKHOAN
    WHERE MATK = @MATK
COMMIT TRAN
```

39

39

## Mức cô lập

### ❖ Repeatable Read

- Đặc điểm: Phát S khi đọc và giữ S đến khi transaction kết thúc.
- Lỗi:
  - Bóng ma
  - Mất dữ liệu cập nhật
- Giải quyết: Giải quyết được **Dirty Read** và **Unrepeatable Read**. Không ngăn chặn lệnh insert dữ liệu thỏa điều kiện thiết lập S do đó không giải quyết được **phantom**.

40

40

## Mức cô lập

<b>T<sub>1</sub></b>	<b>T<sub>2</sub></b>
<b>Set → Repeatable Read</b>	
<b>Select (TAIKHOAN)</b> <b>T<sub>1</sub> : Xin khóa S</b> <b>SQL : Cấp khóa S</b> <b>T<sub>1</sub> : giữ khóa S đến hết giao tác T<sub>1</sub></b>	
	<b>Update (TAIKHOAN)</b> <b>T<sub>2</sub> : Xin khóa X</b> <b>SQL : Không cấp khóa X do T1 đang giữ khóa S</b>
<b>Commit</b>	<b>Commit</b>

41

41

## Mức cô lập

**Mức cô lập được gán trên giao tác T<sub>1</sub> (giao tác đọc dữ liệu)**

```

CREATE PROC USP_DANGNHAP
    @MAKH CHAR(10),
    @MATKHAU CHAR(10)
AS
BEGIN TRAN
    SET TRAN ISOLATION LEVEL REPEATABLE READ
    IF EXISTS (SELECT * FROM KHACHHANG
        WHERE MAKH = @MAKH
        AND MATKHAU = @MATKHAU)
        SELECT *
        FROM KHACHHANG
        WHERE MAKH = @MAKH AND MATKHAU = @MATKHAU
COMMIT TRAN

```

42

42

## Mức cô lập

### ❖ Serializable

- Giống *Repeatable Read*, có ngăn chặn lệnh insert dữ liệu thoả điều kiện thiết lập S. Giải quyết được **Dirty Read**, **Unrepeatable Read** và **Phantom**

43

43

## Mức cô lập

<b>T<sub>1</sub></b>	<b>T<sub>2</sub></b>
Set → Serializable	
<b>Select (TAIKHOAN)</b> <b>T<sub>1</sub> : Xin khóa S</b> <b>SQL : Cập khóa S</b> <b>T<sub>1</sub> : giữ khóa S đến hết giao tác</b> <b>Ngăn chèn dữ liệu vào tập đang khóa</b>	
	<b>Insert (TAIKHOAN)</b> <b>T<sub>2</sub> : Xin khóa X</b> <b>SQL : Không cấp khóa X do T<sub>1</sub></b> <b>ngăn không cho insert vào</b>
Commit	Commit

44

44

## Mức cô lập

	Lost Update	Dirty Read	Unrepeatable Read	Phantom Read
<b>Read Uncommitted</b>	☒	☒	☒	☒
<b>Read Committed</b>	☒	☑	☒	☒
<b>Repeatable Read</b>	☑ (deadlock)	☑	☑	☒
<b>Serializable</b>	☑ (deadlock)	☑	☑	☑

☑ **Giải quyết được**

☒ **Không giải quyết được**

45

45

## Cơ chế khóa

STT	Khóa	Ý nghĩa
1	<b>ReadUncommitted / NoLock</b>	Không thiết lập shared lock khi đọc (tương tự như mức cô lập Read Uncommitted)
2	<b>ReadCommitted</b>	-Đây là chế độ mặc định (tương tự readcommitted) -Chỉ đọc những dữ liệu đã commit -Thiết lập shared lock trên đơn vị dữ liệu đọc và mở clock ngay khi đọc xong.
3	<b>RepeatableRead</b>	Thiết lập shared lock khi đọc và giữ shared lock đến hết giao tác

<https://learn.microsoft.com/en-us/sql/t-sql/queries/hints-transact-sql-table?view=sql-server-ver16>

46

46

## Cơ chế khóa

STT	Khóa	Ý nghĩa
4	<b>Serializable / Holdlock</b>	-Thiết lập shared lock khi đọc, giữ lock đến hết giao tác. -Không cho insert dòng thỏa điều kiện thiết lập khóa
5	<b>Updlock</b>	-Dùng updatelock thay cho shared lock. -Chỉ sử dụng trong câu select. -Uplock được giữ đến hết giao tác.
6	<b>XLock</b>	Chỉ định dùng khóa độc quyền
7	<b>Readpast</b>	-Chỉ khóa dòng dữ liệu đang thao tác. -Áp dụng cho câu lệnh select. -Chỉ dùng được với READ COMMITTED hoặc REPEATABLE READ

47

47

## Cơ chế khóa

STT	Khóa	Ý nghĩa
8	<b>RowLock</b>	Chỉ đặt khóa trên dòng cần thao tác
9	<b>TabLock</b>	-Khóa toàn bộ dòng trên bảng đang thao tác. -Các thao tác (Insert / Update / Delete) không thể thực hiện trên bảng này.
10	<b>TabLockX</b>	Xlock + TabLock

- ⇒ 1, 2, 3, 5, 6, 7 chỉ có ý nghĩa khi dùng trong câu select.  
 ⇒ 1, 2, 3, 5, 6, 7 có thể kết hợp với 4 (khóa theo kiểu key-range) và 8, 9 (chỉ ra đơn vị cần khóa)

48

48

## Thiết lập khóa

Select ...

From {Tab1 Alias1 with (*Lock\_mode* [...n])} [...n])

Where ...

Ví dụ:

Select SV.HoVaTen, K.TenKhoa

From SinhVien SV with (ReadCommitted),

Khoa K with (Updlock)

Where SV.Khoa = K.Ma

49

49

## Deadlock

☞ Conversion deadlock

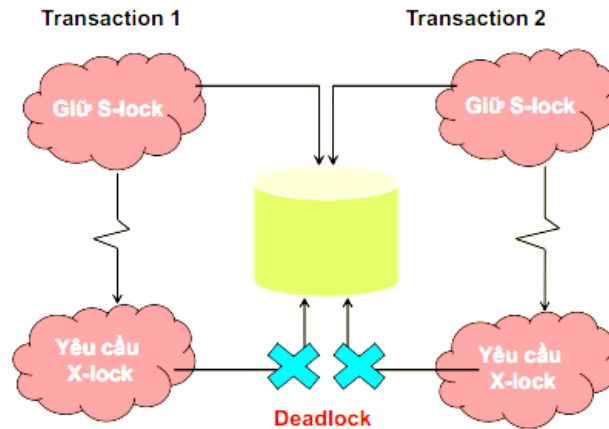
☞ Cycle deadlock

50

50



# Conversion deadlock



51

51

# Conversion deadlock

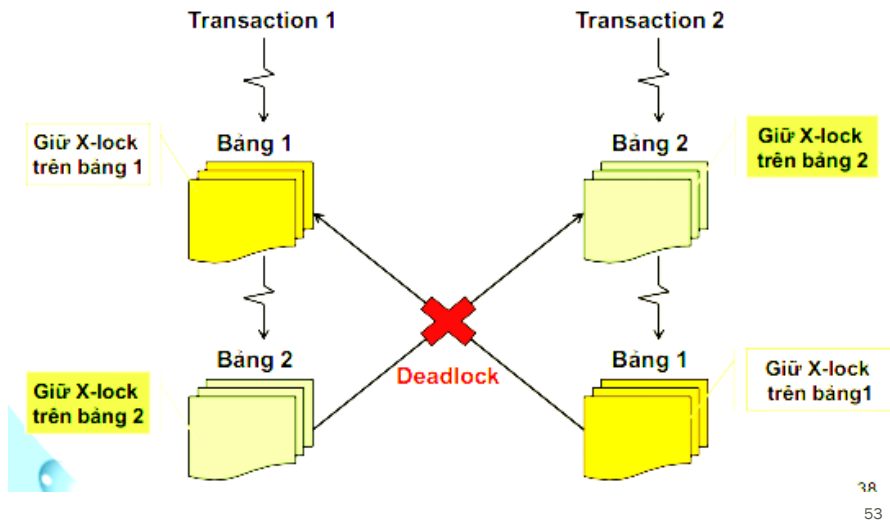
T1	T2
Begin tran <b>SET TRAN ISOLATION LEVEL SERIALIZABLE</b> 1 Read (A) Waitfor delay '0:0:15' 2 Write (A) Commit	Begin tran <b>SET TRAN ISOLATION LEVEL SERIALIZABLE</b> 3 Read (A) 4 Write (A) Commit

**T1 chờ T2 trả khóa đọc trên A, và T2 chờ T1 trả khóa đọc trên A => Hệ thống bị treo**

52

52

# Cycle deadlock



53

# Cycle deadlock

T1	T2
Begin tran	Begin tran
1 Write (A) Waitfor delay '0:0:15'	3 Write (B)
2 Write (B)	4 Write (A)
Commit	Commit

**T1 chờ T2 trả khóa ghi trên B, và T2 chờ T1 trả khóa ghi trên A Hệ thống bị treo**

54

54

## Giải pháp xử lý Deadlock

⇒ Chuyển về dạng tuần tự - phối hợp transaction, mức độ cô lập và lock

55

55