

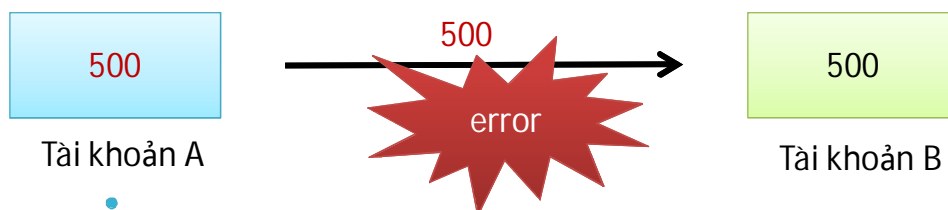
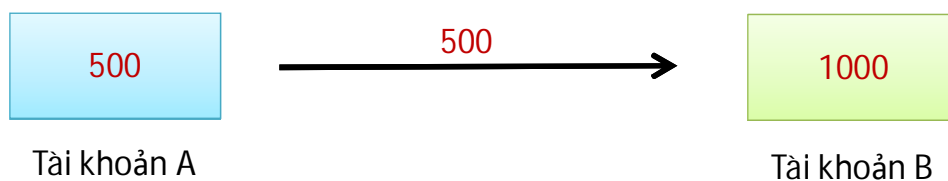
Seminar

Transaction & Concurrency Control

Hoàng Anh Tú

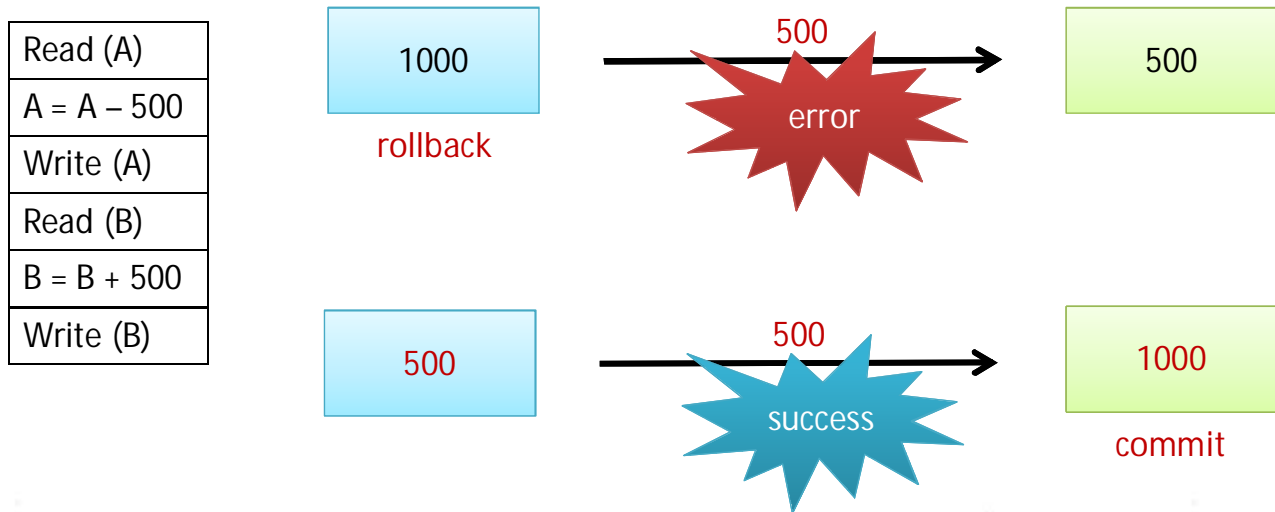
1

Chuyển tiền ngân hàng



Giao tác (Transaction)

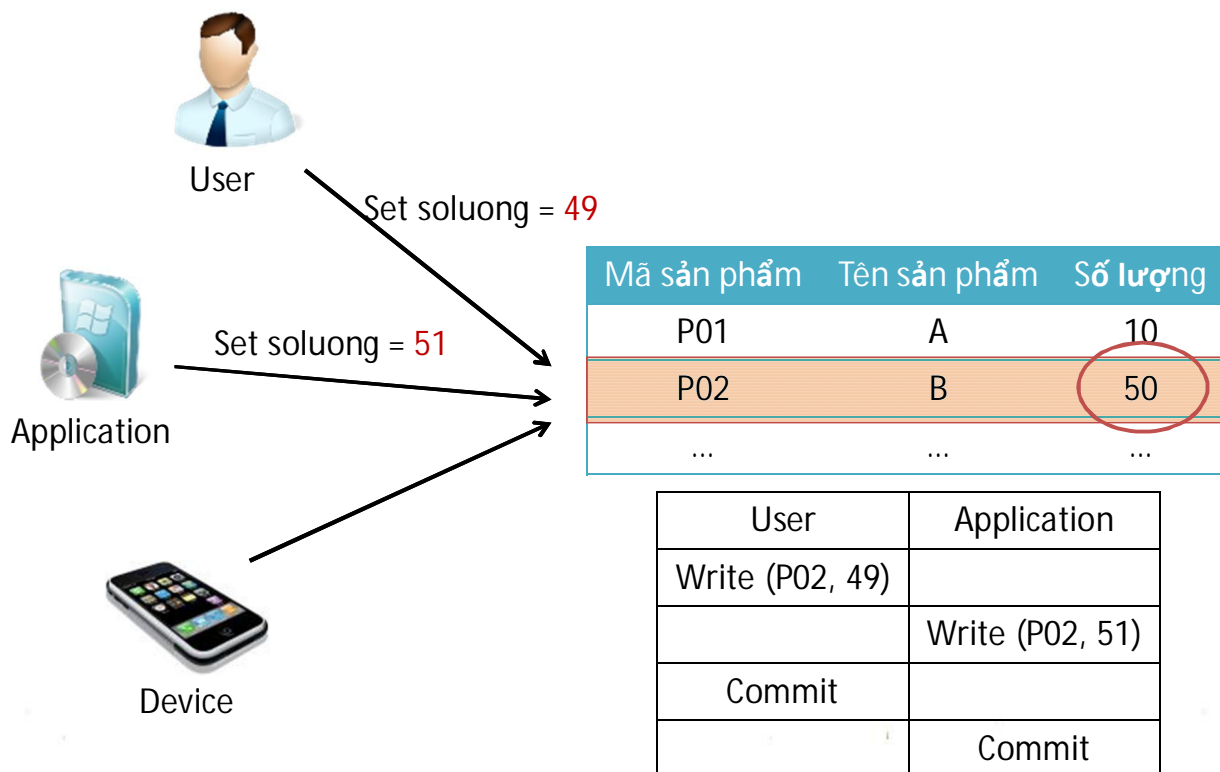
- Bao gồm nhiều lệnh SQL nhưng khi thực thi chỉ được xem xét như 1 lệnh.



Hoàng Anh Tú

3

Giao tác có thể thực thi cùng lúc

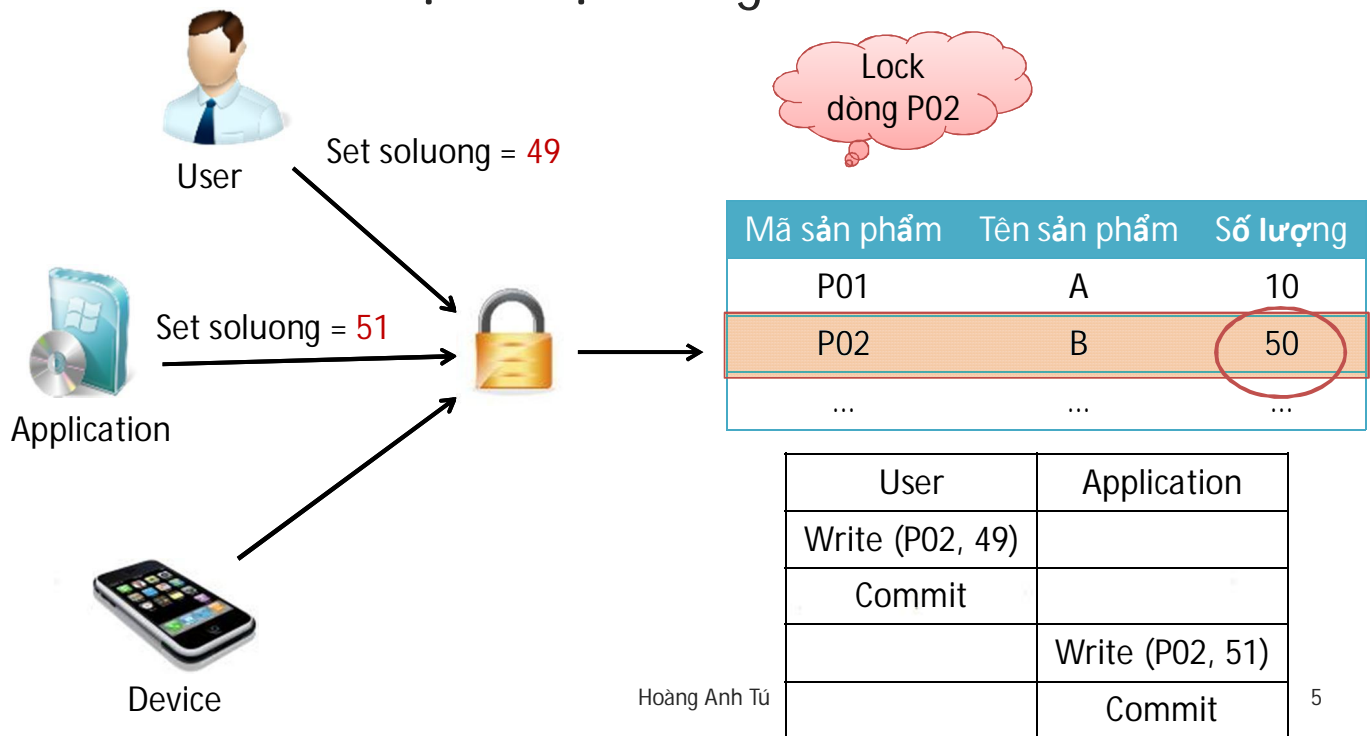


Hoàng Anh Tú

4

Điều khiển truy xuất đồng thời (Concurrency Control)

- Xử lý cho phép nhiều ứng dụng cùng thao tác lên 1 đơn vị dữ liệu cùng lúc



Content





Concurrency Control Problems

Các lỗi phát sinh khi thực thi nhiều giao tác cùng lúc

Hoàng Anh Tú

7

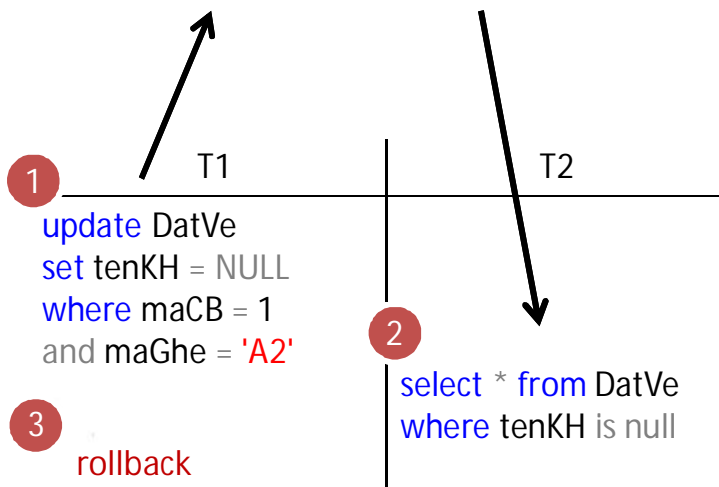
Các vấn đề khi xử lý đồng thời

- Các giao tác thực hiện đồng thời có khả năng làm vi phạm tính nhất quán của dữ liệu

Giao tác 1	Giao tác 2	Nhận xét
Đọc	Đọc	Không xảy ra tranh chấp
Đọc	Viết	Có tranh chấp xảy ra
Viết	Đọc	Có tranh chấp xảy ra
Viết	Viết	Cùng lúc chỉ có 1 giao tác được ghi trên 1 đơn vị dữ liệu

Dirty read (Uncommitted read)

maCB	maGhe	tenKH
1	A1	HATu
1	A2	TVAn
1	A3	HTTam



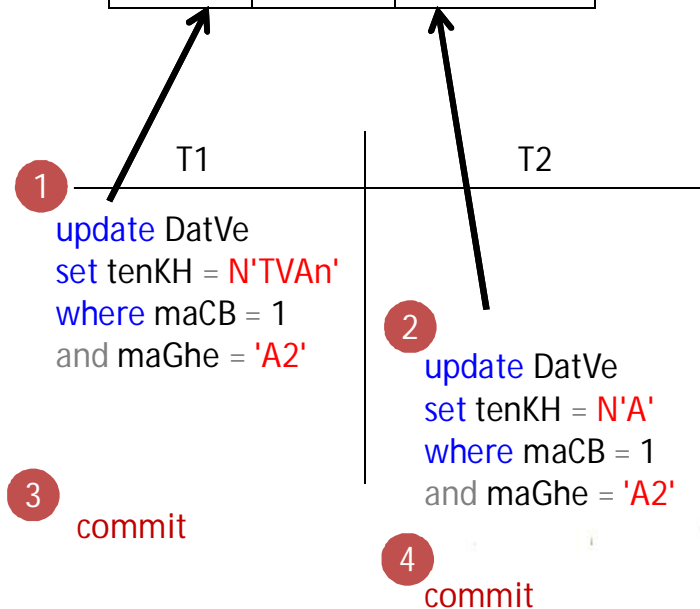
maCB	maGhe	tenKH
1	A2	NULL

Hoàng Anh Tú

9

Lost update

maCB	maGhe	tenKH
1	A1	HATu
1	A2	NULL
1	A3	HTTam



Hoàng Anh Tú

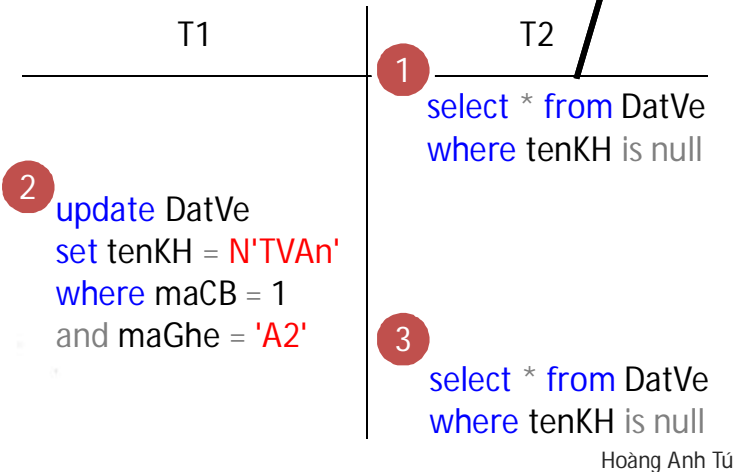
10

Unrepeatable read

maCB	maGhe	tenKH
1	A1	HATu
1	A2	NULL
1	A3	HTTam

maCB	maGhe	tenKH
1	A2	NULL

maCB	maGhe	tenKH
------	-------	-------



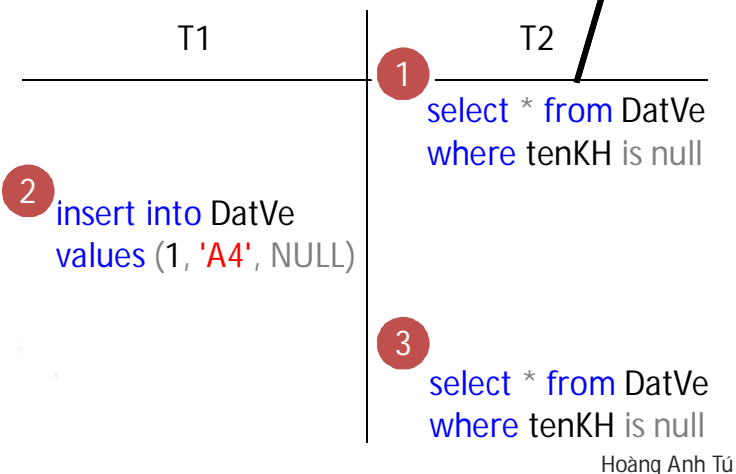
11

Phantom

maCB	maGhe	tenKH
1	A1	HATu
1	A2	TVAn
1	A3	HTTam

maCB	maGhe	tenKH
------	-------	-------

maCB	maGhe	tenKH
1	A4	NULL



12



Concurrency Control in SQL Server 2008

Điều khiển truy xuất đồng thời trong Cơ sở dữ liệu bằng SQL Server

Hoàng Anh Tú

13

Transaction Management

- Implicit Transaction: các transaction được tự động bắt đầu khi thực hiện 1 số thao tác cơ sở dữ liệu: *INSERT, UPDATE, DELETE,...*
 - Hoạt động trong chế độ *auto-commit*
- Explicit Transaction: các transaction phải khai báo tường minh

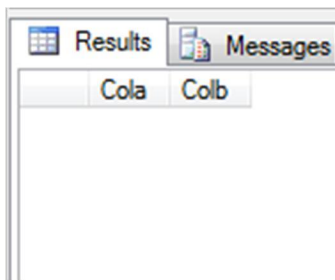
Auto-commit

Cơ chế quản lý Transaction mặc định của SQL Server

```
CREATE TABLE TestBatch (Cola INT PRIMARY KEY, Colb CHAR(3))
GO
INSERT INTO TestBatch VALUES (1, 'aaa')
INSERT INTO TestBatch VALUES (2, 'bbb')
INSERT INTO TestBatch VALUSE (3, 'ccc') /* Syntax error */
GO
SELECT * FROM TestBatch
```

Mỗi lệnh là 1 transaction

Không trả về dòng nào



Explicit Transaction

Khai báo tường minh Transaction

BEGIN TRAN

```
update DatVe
set tenKH = NULL
where maCB = 1
and maGhe = 'A2'
```

ROLLBACK

Đặt tên cho Transaction

BEGIN TRAN T1

```
update DatVe
set tenKH = NULL
where maCB = 1
and maGhe = 'A2'
```

ROLLBACK TRAN T1

Nested Transaction

```
begin tran T1  
  insert into DatVe  
  values (1, 'A6', NULL)
```

```
begin tran T2  
  insert into DatVe  
  values (1, 'A7', NULL)
```

```
rollback tran T2  
commit tran T1
```

maCB	maGhe	tenKH
1	A1	HATu
1	A2	TVAn
1	A3	HTTam



maCB	maGhe	tenKH
1	A1	HATu
1	A2	TVAn
1	A3	HTTam
1	A6	NULL
1	A7	NULL

Msg 6401, Level 16, State 1, Line 9
Cannot roll back T2. No transaction or savepoint of that name
was found.

Hoàng Anh Tú

17

Controlling Lock

- Điều khiển truy xuất đồng thời được thực hiện qua cơ chế lock
- Điều khiển các lock bằng các cơ chế:
 - Database Configuration: cho phép thiết lập database read-only
 - Locking Hints: cho phép xin lock khi thực thi 1 câu lệnh
 - Transaction Isolation Settings: thiết lập mức độ cô lập của transaction

Hoàng Anh Tú

18

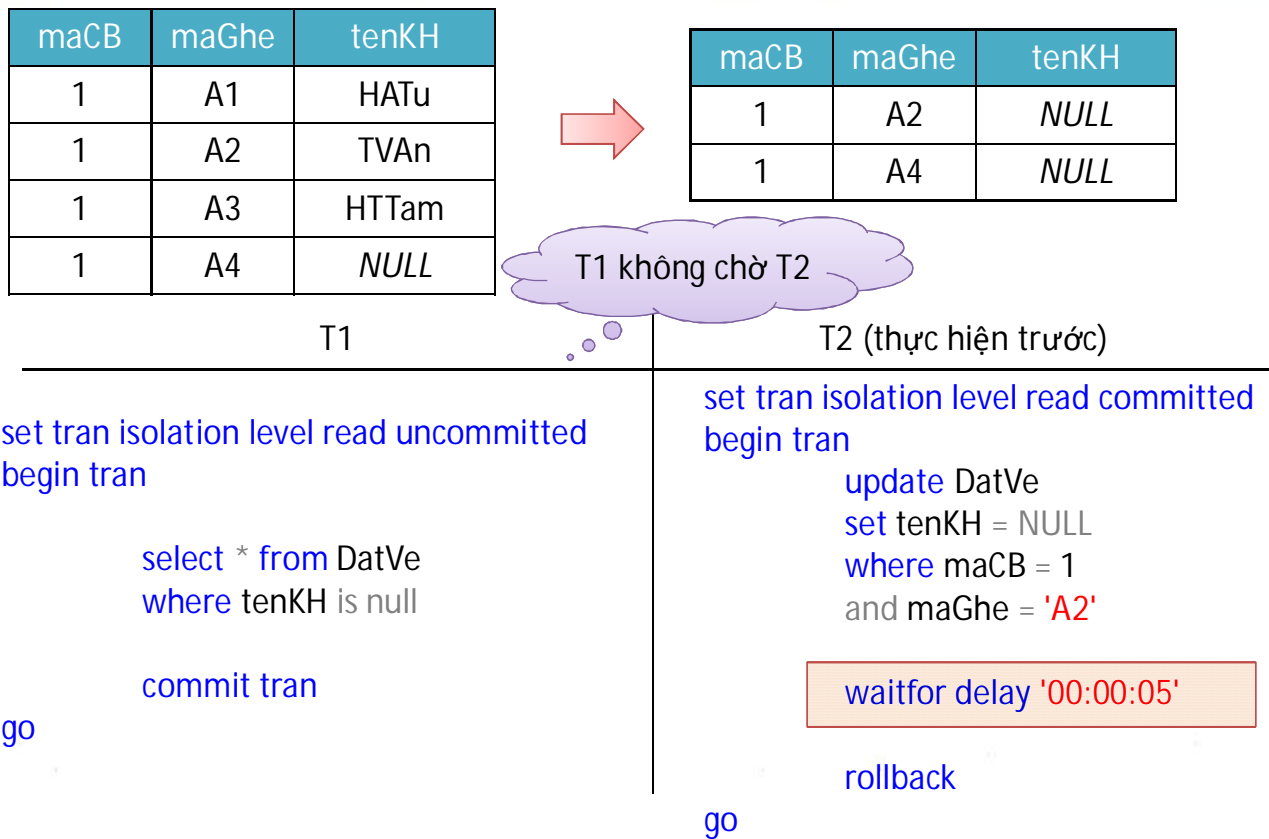
Transaction Isolation Settings

- Các transaction có tính chất ACID: [Atomicity, Consistency, Isolation, Durability](#)
- *Transaction Isolation Settings* cho phép thiết lập lại mức cô lập của transaction
 - Read Uncommitted
 - Read Committed
 - Repeatable Read
 - Serializable

Read Uncommitted

- Không thiết lập [*Shared Lock*](#) trên những đơn vị dữ liệu cần đọc.
- Vẫn tạo [*Exclusive Lock*](#) trên đơn vị dữ liệu được ghi, [*Exclusive Lock*](#) được giữ cho đến hết giao tác

Read Uncommitted



Hoàng Anh Tú

21

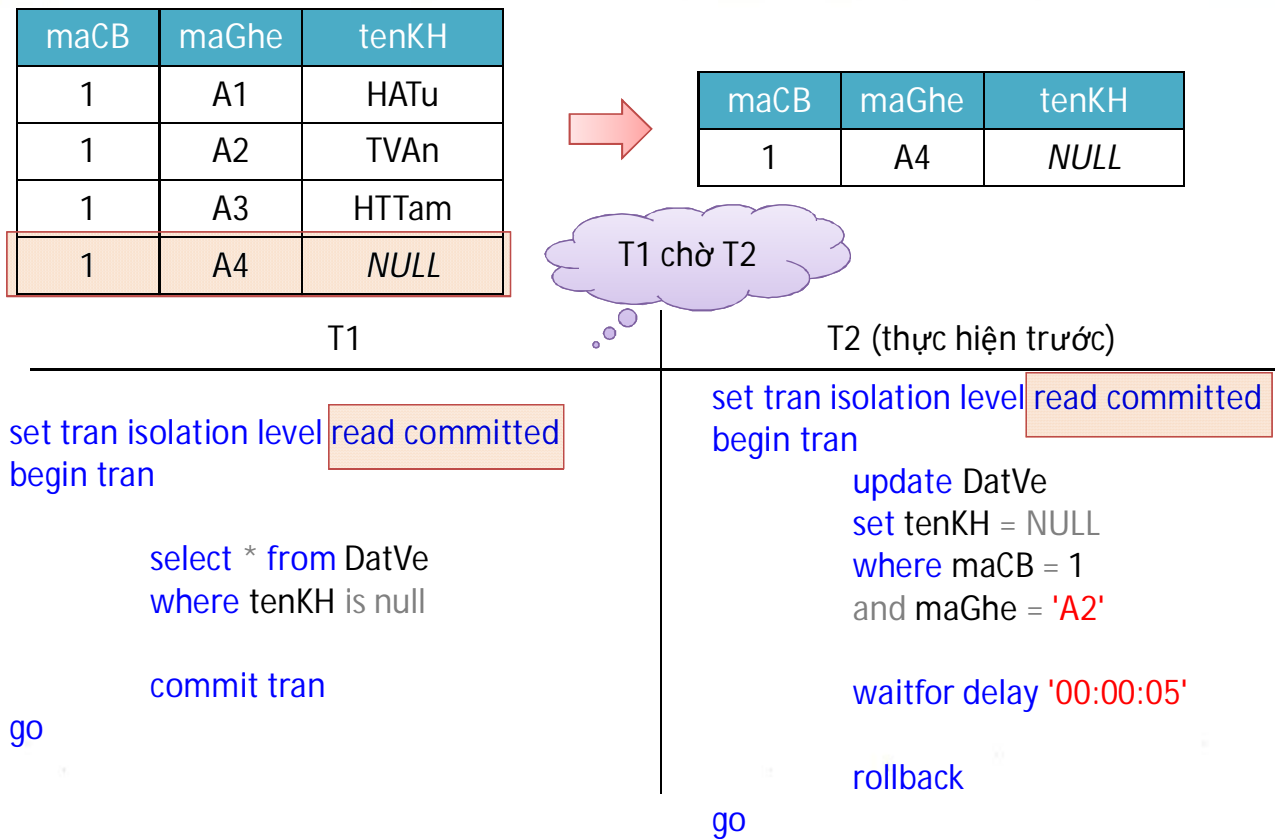
Read Committed

- Mức độ cô lập mặc định của SQL Server
- Tạo *Shared Lock* trên đơn vị dữ liệu được đọc, *Shared Lock* được giải phóng ngay sau khi đọc xong dữ liệu
- Tạo *Exclusive Lock* trên đơn vị dữ liệu được ghi, *Exclusive Lock* được giữ cho đến hết giao tác

Hoàng Anh Tú

22

Giải quyết Dirty Read



Hoàng Anh Tú

23

Repeatable Read

- Tạo *Shared Lock* trên đơn vị dữ liệu được đọc và giữ *shared lock* này đến hết giao tác
- Tạo *Exclusive Lock* trên đơn vị dữ liệu được ghi, *Exclusive Lock* được giữ cho đến hết giao tác

Giải quyết Unrepeatable Read

maCB	maGhe	tenKH
1	A1	HATu
1	A2	NULL
1	A3	HTTam



maCB	maGhe	tenKH
1	A2	NULL

maCB	maGhe	tenKH
1	A2	NULL

T1 (thực hiện trước)

T2

set tran isolation level repeatable read
begin tran

select * from DatVe
where tenKH is null

waitfor delay '00:00:05'

select * from DatVe
where tenKH is null

commit

go

set tran isolation level read committed
begin tran

update DatVe
set tenKH = N'TVAn'
where maCB = 1
and maGhe = 'A2'

commit

go

Hoàng Anh Tú

25

Serializable

- Tạo *Shared Lock* trên đơn vị dữ liệu được đọc và giữ *shared lock* này đến hết giao tác
- Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập *Shared Lock*
- Tạo *Exclusive Lock* trên đơn vị dữ liệu được ghi, *Exclusive Lock* được giữ cho đến hết giao tác

Giải quyết Phantom

maCB	maGhe	tenKH
1	A1	HATu
1	A2	TVAn
1	A3	HTTam



maCB	maGhe	tenKH
------	-------	-------

maCB	maGhe	tenKH
------	-------	-------

T1 (thực hiện trước)

```
set tran isolation level serializable
begin tran
```

```
select * from DatVe
where tenKH is null
```

```
waitfor delay '00:00:05'
```

```
select * from DatVe
where tenKH is null
```

```
commit
```

go

T2

```
set tran isolation level read committed
begin tran
```

```
insert into DatVe
values (1, 'A4', NULL)
```

```
commit
```

go

Hoàng Anh Tú

27

Deadlock

- Trong xử lý đồng thời, giao tác này phải chờ giao tác khác
- Điều gì xảy ra nếu 2 giao tác phải chờ lẫn nhau

Phân loại deadlock

- Gồm 2 loại:
 - Cyclic deadlock
 - Conversion deadlock

CYCLIC DEADLOCK

T1	T2
insert/update/delete(A)	
	insert/update/delete(B)
insert/update/delete(B)	
	insert/update/delete(A)
commit	commit

T1	T2
select(A)	
	select(A)
insert/update/delete(A)	
	insert/update/delete(A)
commit	commit



Concurrency Control in ADO.NET

Điều khiển truy xuất đồng thời ở mức ứng dụng bằng ADO.NET

Connect to database

```
SqlConnection con = new  
    SqlConnection("data source=127.0.0.1;"  
        + "initial catalog=QLTV;"           // Địa chỉ server  
        + "user=sa;"                       // Tên cơ sở dữ liệu  
        + "password=123");                 // Tên tài khoản đăng nhập  
                                           // Tài khoản
```

Calling Stored Procedure

```
string strCmd = "sp_ThongTinDocGia";  
SqlCommand cmd = new SqlCommand(strCmd, con);  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.Parameters.Add(new SqlParameter("@madocgia", 1));  
SqlDataAdapter adapt = new SqlDataAdapter(cmd);  
DataSet dbset = new DataSet();  
adapt.Fill(dbset);
```

Gọi stored
procedure

Truyền tham
số

Lấy dữ liệu

Transaction in ADO.NET

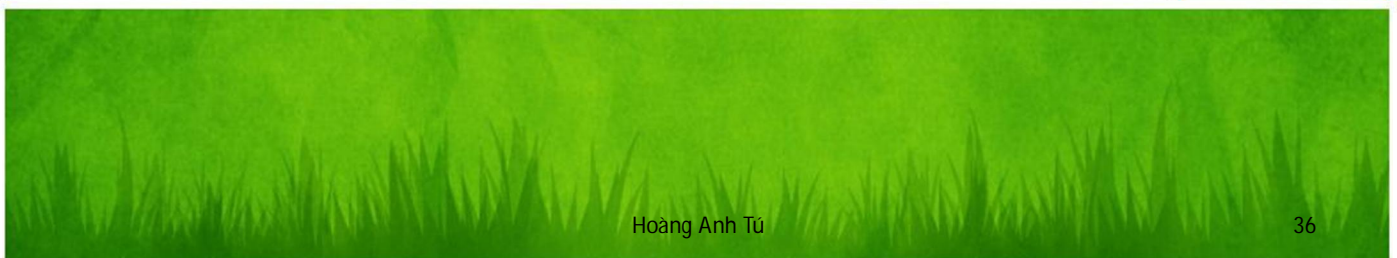
```
try{  
    SqlConnection transaction = con.BeginTransaction();  
    SqlCommand cmd = new SqlCommand(strCmd, con);  
    cmd.CommandType = CommandType.StoredProcedure;  
    cmd.Parameters.Add(new SqlParameter("@madocgia", 1));  
    cmd.Transaction = transaction;  
    SqlDataAdapter adapt = new SqlDataAdapter(cmd);  
  
    adapt.Fill(dbset);  
    transaction.Commit();  
}catch{  
    transaction.Rollback();  
}
```

Hoàng Anh Tú

35



Thanks for your
attention



Hoàng Anh Tú

36