

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



HỒ PHÚC LÂM
MSSV

Báo cáo thực hành: **TRANSACTION VÀ BẢO MẬT PHÂN QUYỀN**

Lớp học phần: HQTCSDL_DHTMDT17.HE.HL

Môn học: Hệ quản trị cơ sở dữ liệu

GVHD: ThS Nguyễn Đức Cường

TP. Hồ Chí Minh, tháng 08 năm 2024

MỤC LỤC

I.ROLE, PERMISSION	1
7)Tạo hai login SQL server Authentication User2 và User3	1
8)Tạo một database user User2 ứng với login User2 và một database user User3 ứng với login User3 trên CSDL NorthWind.	2
9)Tạo 2 kết nối đến server thông qua login User2 và User3, sau đó thực hiện các thao tác truy cập CSDL của 2 user tương ứng (VD: thực hiện câu select), có thực hiện được không?	2
10)Gán quyền select trên Employees cho User2, kiểm tra kết quả. Xóa quyền select trên Employees cho User2. Ngắt 2 kết nối của User2 và User3	3
11)Trở lại kết nối của sa, tạo một user-defined database Role tên employee_role trên CSDL NorthWind, sau đó gán các quyền Select,Update, Delete cho employee_role....	5
12)Thêm các User2 và User3 vào employee_role. Tạo lại 2 kết nối đến server thông qua login User2 và User3 thực hiện các thao tác sau:	6
12a)Tại kết nối với User2, thực hiện câu lệnh select để xem thông tin của bảng Employees	6
12b)Tại kết nối của User3, thực hiện cập nhật Title='Vice President' của nhân viên có EmployeeID=1	7
12c)Tại kết nối User2, dùng câu lệnh select xem lại kết quả.....	8
12d)Xóa role employee_role, (quá trình xóa role ?)	9
I.SINGLE TRANSACTION.....	10
1) Thêm vào bảng Categories một dòng dữ liệu tùy ý bằng câu lệnh INSERT..VALUES.	10
2) Tắt chế độ autocommit của SQL Server (SET IMPLICIT_TRANSACTIONS ON). Tạo đoạn batch gồm các thao tác:	12

- 3) Viết đoạn batch thực hiện các thao tác sau (lưu ý thực hiện lệnh SET XACT_ABORT ON: nếu câu lệnh T-SQL làm phát sinh lỗi run-time, toàn bộ giao dịch được chấm dứt và Rollback)..... 13
- 4) Thực hiện lệnh SET XACT_ABORT OFF (những câu lệnh lỗi sẽ rollback, transaction vẫn tiếp tục) sau đó thực thi lại các thao tác của đoạn batch ở câu 3. Quan sát kết quả và giải thích kết quả?..... 14

II.CONCURRENT TRANSACTIONS..... 15

- 1)Tạo bảng Accounts (AccountID int NOT NULL PRIMARY KEY,balance int NOT NULL CONSTRAINT unloanable_account CHECK (balance >= 0) Và chèn dữ liệu: 15
- 2)SET TRANSACTION ISOLATION LEVEL 16
- 3)Mở 2 cửa sổ Query của SQL server, thiết lập SET TRANSACTION ISOLATION LEVEL READ COMMITTED ở cả 2 cửa sổ (tạm gọi là clients A bên trái, và client B bên phải). 18
- 5)Giả sử có 2 giao dịch chuyển tiền từ tài khoản 101 và 202 như sau:..... 24
- 6)Xóa tất cả dữ liệu của bảng Accounts. Thêm lại các dòng mới INSERT INTO Accounts (AccountID ,balance) VALUES (101,1000); INSERT INTO Accounts (AccountID ,balance) VALUES (202,2000); 26
- 7)Xóa tất cả dữ liệu của bảng Account, thêm lại các dòng mới..... 29

I.ROLE, PERMISSION

7)Tạo hai login SQL server Authentication User2 và User3

```
/*7)Tạo hai login SQL server Authentication User2 và User3*/  
create login User2 with password = 'a@1'  
create login User3 with password = 'a@1'  
Go  
  
--xem thông tin  
select * from SYS.SQL_LOGINS  
Go
```

100 %

Results Messages

	name	principal...	sid	ty...	type_desc	is_disabl...	create_date
1	sa	1	0x01	S	SQL_LOGIN	0	2003-04-08 09:10:35.460
2	##MS_PolicyEventProcessingLogin##	256	0x21B28D920291094FBEC06E0F55CF27C6	S	SQL_LOGIN	1	2012-02-10 21:07:46.267
3	##MS_PolicyTsqlExecutionLogin##	257	0xFB5428192C68DE479445435923D3CE58	S	SQL_LOGIN	1	2012-02-10 21:07:46.637
4	User2	266	0x1AA02D1B87DD9B4B8E5641AB0824B...	S	SQL_LOGIN	0	2024-08-02 16:47:39.660
5	User3	267	0xB99C55FDA44AEF4E845133E898D2B9...	S	SQL_LOGIN	0	2024-08-02 16:47:39.667

8) Tạo một database user User2 ứng với login User2 và một database user User3 ứng với login User3 trên CSDL NorthWind.

```

/*8) Tạo một database user User2 ứng với login User2 và một database
user User3 ứng với login User3 trên CSDL NorthWind.*/
use Northwind
Go

--Tạo database user 2 và user 3
create user User2 for login User2
create user User3 for login User3
Go

--Xem lại các user
select * from sys.database_principals where type_desc = 'SQL_USER';

```

name	principal_id	type	type_desc	default_schema_name	create_date	modify_date	owning_principal_id	sid	is_fixed_role	authentication
dbo	1	S	SQL_USER	dbo	2003-04-08 09:10:42.287	2024-06-25 13:40:31.537	NULL	0x01	0	1
guest	2	S	SQL_USER	guest	2003-04-08 09:10:42.317	2003-04-08 09:10:42.317	NULL	0x00	0	0
INFORMATION_SCHEMA	3	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL	0	0
sys	4	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL	0	0
User2	5	S	SQL_USER	dbo	2024-08-02 16:55:49.780	2024-08-02 16:55:49.780	NULL	0x1AA02D1B87DD9B48BE5641AB0824BC35	0	1
User3	6	S	SQL_USER	dbo	2024-08-02 16:55:49.783	2024-08-02 16:55:49.783	NULL	0xB99C55FDA44AEF4E845133E898D2B98A	0	1

9) Tạo 2 kết nối đến server thông qua login User2 và User3, sau đó thực hiện các thao tác truy cập CSDL của 2 user tương ứng (VD: thực hiện câu select), có thực hiện được không?

--Kết nối với User 2 và thực hiện lệnh select

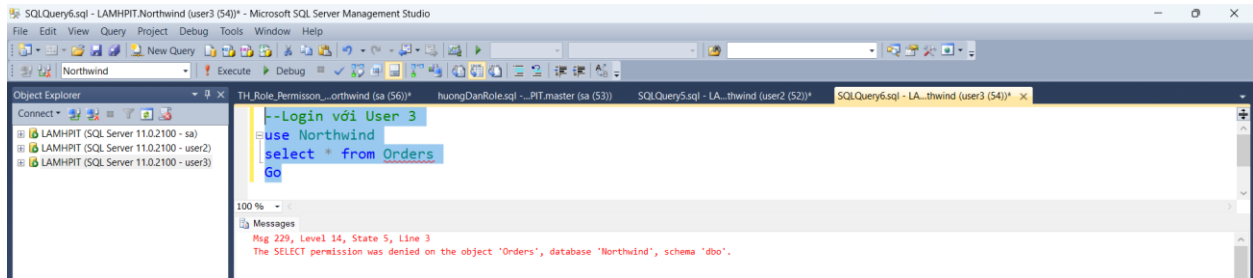
```

--Login với User 2
use Northwind
select * from Orders
Go

```

Msg 229, Level 14, State 5, Line 3
The SELECT permission was denied on the object 'Orders', database 'Northwind', schema 'dbo'.

--Kết nối với User 3 và thực hiện lệnh select

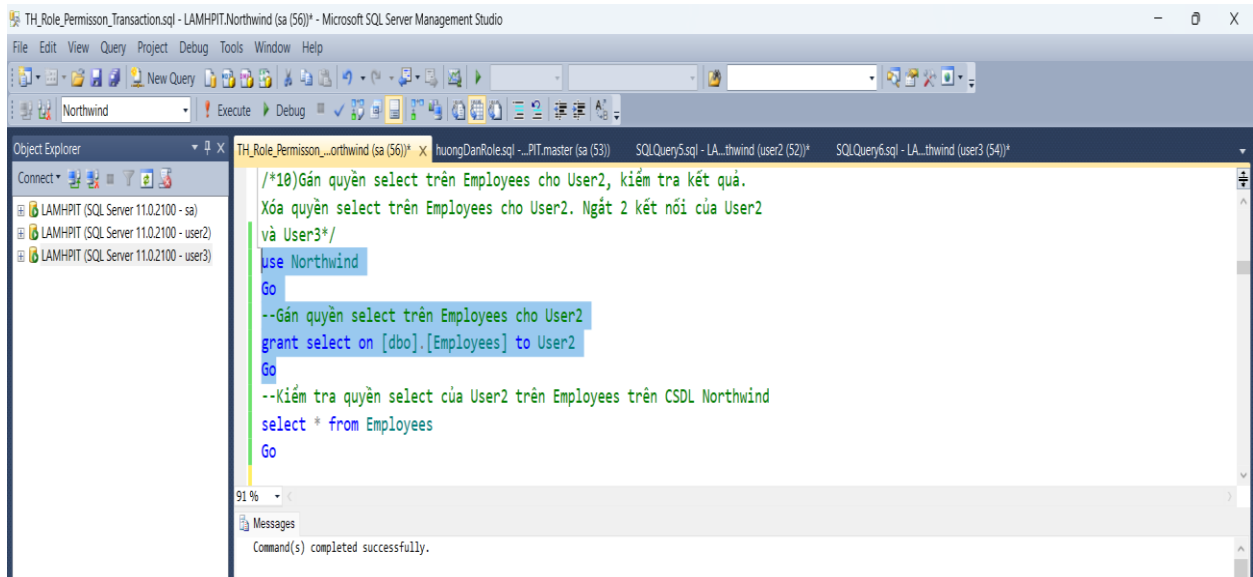


Kết quả:

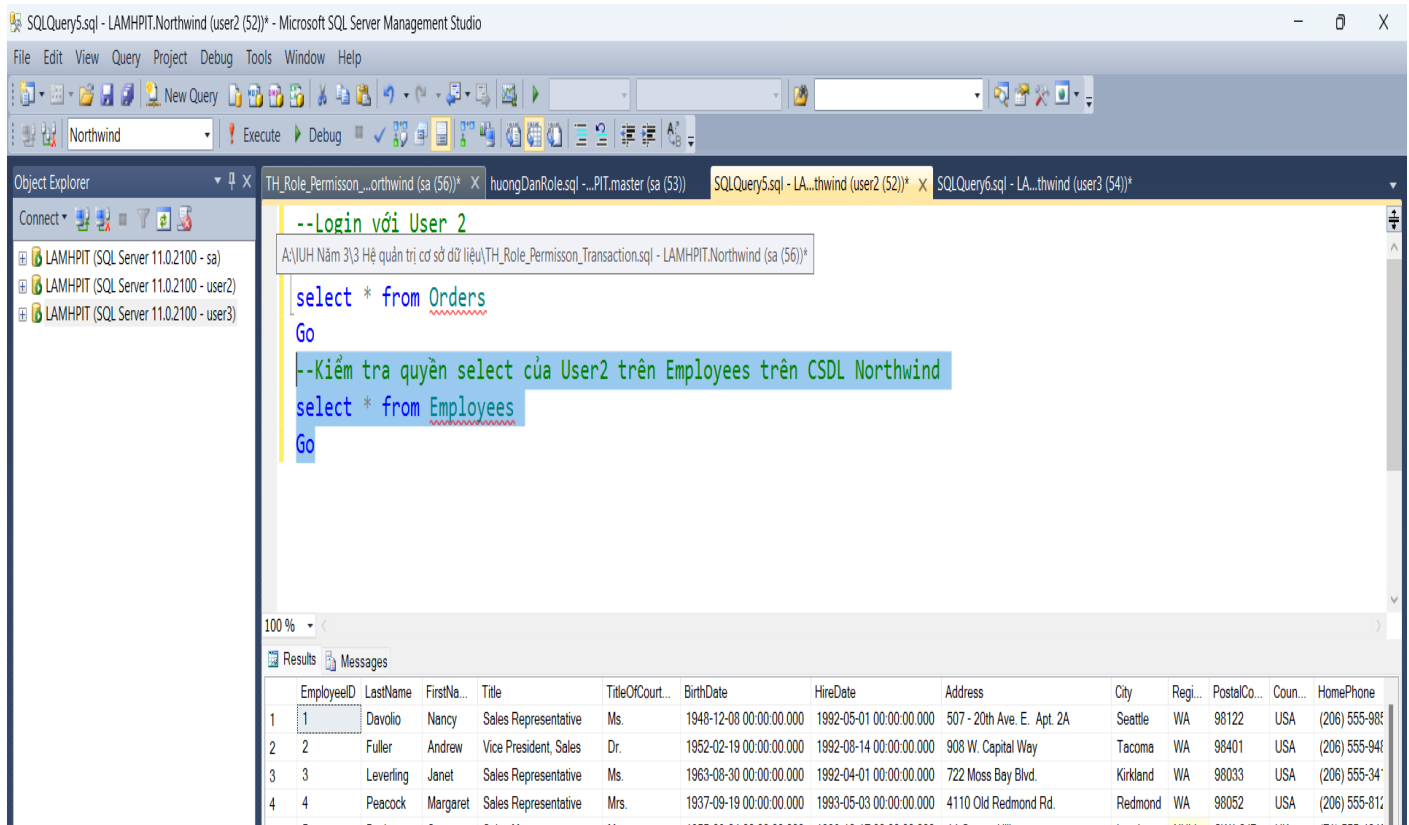
*/*Do mới tạo login và user 2 và 3 mà chưa gán quyền nên bị lỗi không xem được các bảng trong CSDL Northwind*/*

10)Gán quyền select trên Employees cho User2, kiểm tra kết quả. Xóa quyền select trên Employees cho User2. Ngắt 2 kết nối của User2 và User3

--Gán quyền select trên Employees cho User2



--Kiểm tra quyền select của User2 trên Employees trên CSDL Northwind



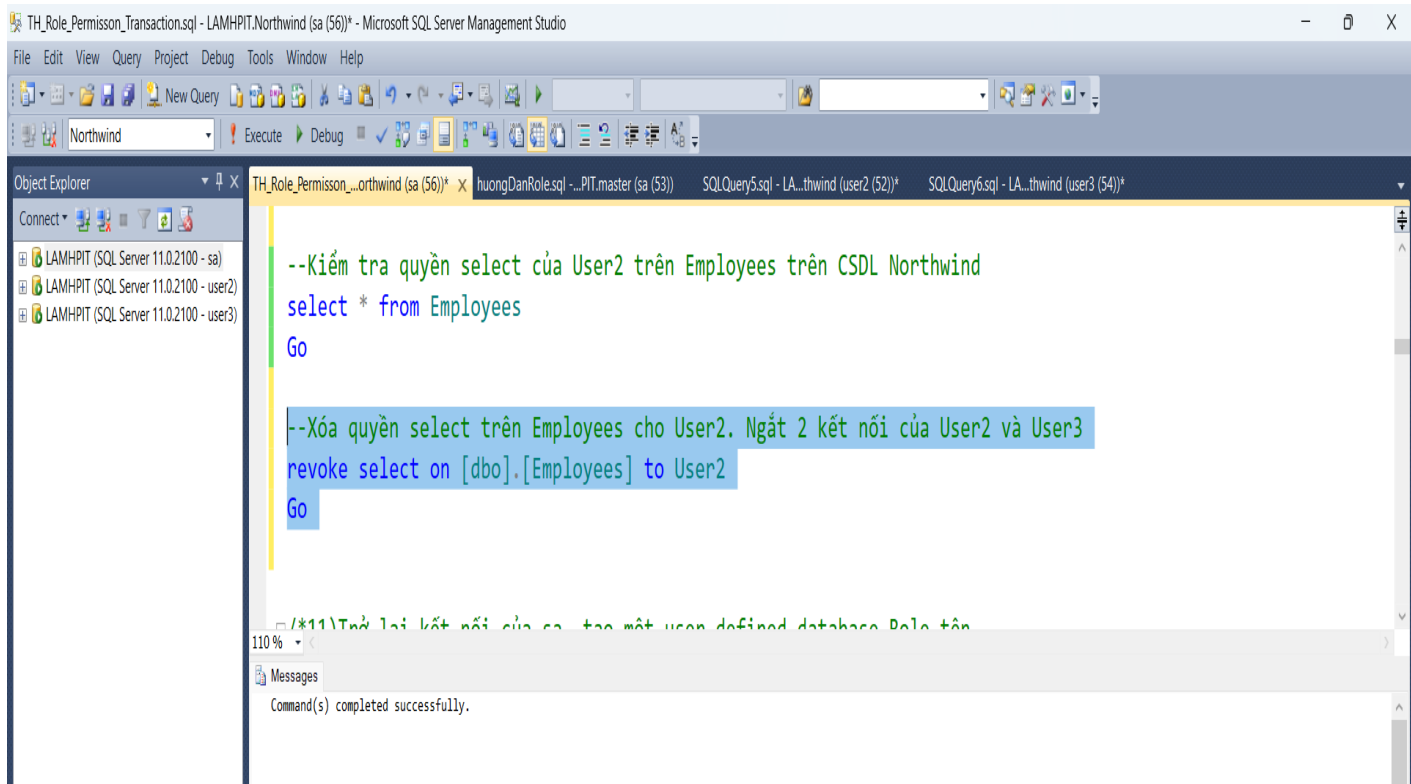
The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the active query is 'SQLQuery5.sql - LAMHPIT.Northwind (user2 (52))*'. The Object Explorer on the left shows the 'Northwind' database selected. The query editor contains the following SQL code:

```
--Login với User 2
select * from Orders
Go
--Kiểm tra quyền select của User2 trên Employees trên CSDL Northwind
select * from Employees
Go
```

The 'Results' tab at the bottom displays the output of the 'select * from Employees' query. The results are shown in a table with the following columns: EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate, Address, City, Region, PostalCode, Country, and HomePhone. The first four rows of data are visible:

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone
1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122	USA	(206) 555-9876
2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000	908 W. Capital Way	Tacoma	WA	98401	USA	(206) 555-9402
3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000	722 Moss Bay Blvd.	Kirkland	WA	98033	USA	(206) 555-3443
4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000	4110 Old Redmond Rd.	Redmond	WA	98052	USA	(206) 555-8142

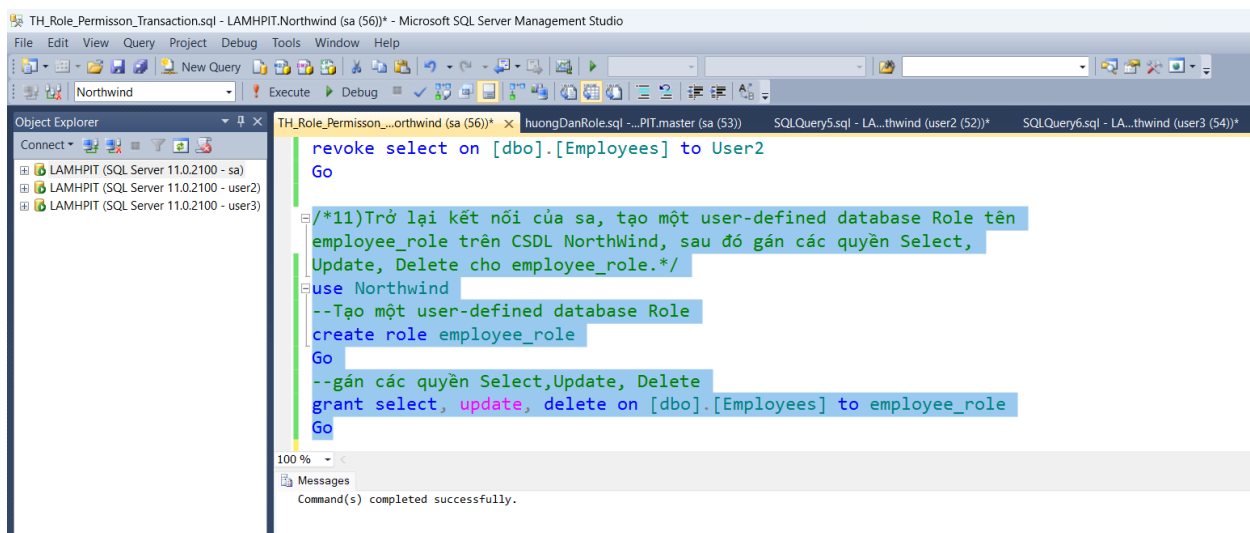
--Xóa quyền select trên Employees cho User2. Ngắt 2 kết nối của User2 và User3



11)Trở lại kết nối của sa, tạo một user-defined database Role tên employee_role trên CSDL NorthWind, sau đó gán các quyền Select,Update, Delete cho employee_role.

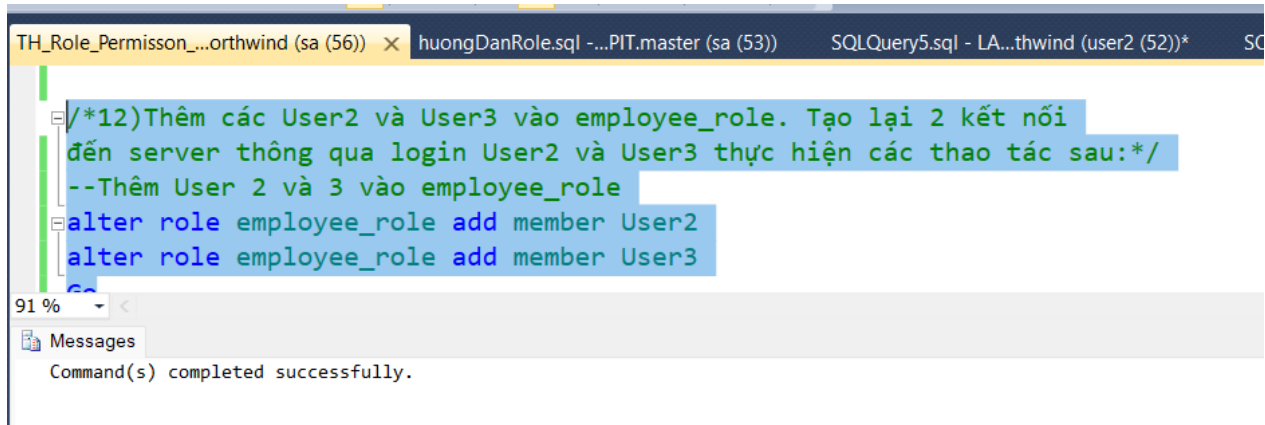
--Tạo một user-defined database Role

--gán các quyền Select,Update, Delete



12)Thêm các User2 và User3 vào employee_role. Tạo lại 2 kết nối đến server thông qua login User2 và User3 thực hiện các thao tác sau:

--Thêm User 2 và 3 vào employee_role



```

/*12)Thêm các User2 và User3 vào employee_role. Tạo lại 2 kết nối
đến server thông qua login User2 và User3 thực hiện các thao tác sau:*/
--Thêm User 2 và 3 vào employee_role
alter role employee_role add member User2
alter role employee_role add member User3

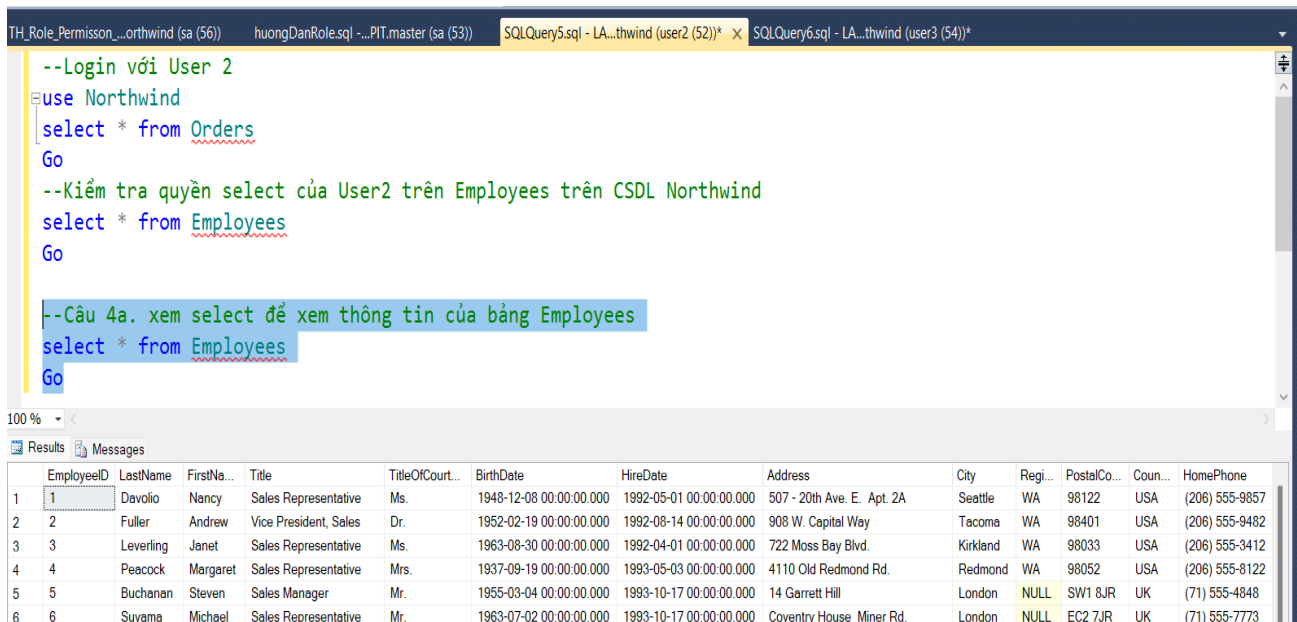
```

91 %

Messages

Command(s) completed successfully.

12a)Tại kết nối với User2, thực hiện câu lệnh select để xem thông tin của bảng Employees



```

--Login với User 2
use Northwind
select * from Orders
Go
--Kiểm tra quyền select của User2 trên Employees trên CSDL Northwind
select * from Employees
Go

--Câu 4a. xem select để xem thông tin của bảng Employees
select * from Employees
Go

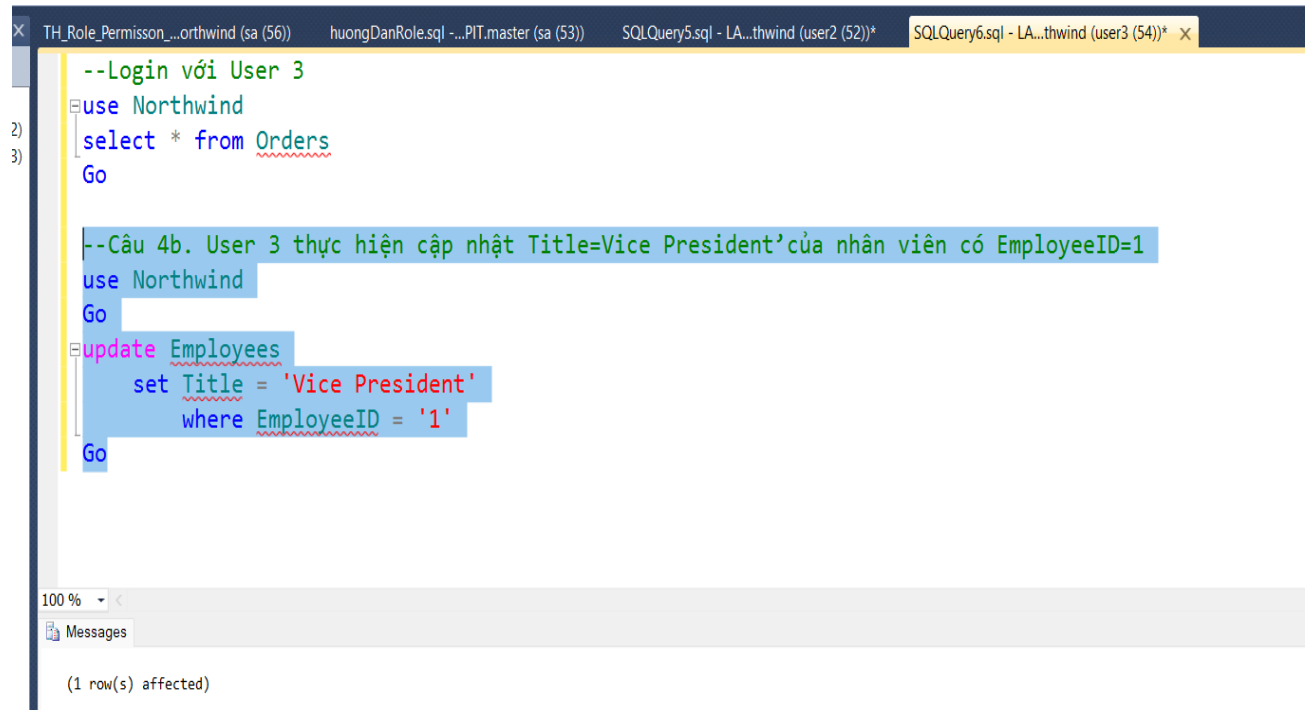
```

100 %

Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate	Address	City	Regi...	PostalCo...	Coun...	HomePhone
1	1	Devolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122	USA	(206) 555-9857
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000	908 W. Capital Way	Tacoma	WA	98401	USA	(206) 555-9482
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000	722 Moss Bay Blvd.	Kirkland	WA	98033	USA	(206) 555-3412
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000	4110 Old Redmond Rd.	Redmond	WA	98052	USA	(206) 555-8122
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17 00:00:00.000	14 Garrett Hill	London	NULL	SW1 8JR	UK	(71) 555-4848
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000	Coventry House Miner Rd.	London	NULL	EC2 7JR	UK	(71) 555-7773

12b) Tại kết nối của User3, thực hiện cập nhật Title=Vice President' của nhân viên có EmployeeID=1



```
--Login với User 3
use Northwind
select * from Orders
Go

--Câu 4b. User 3 thực hiện cập nhật Title=Vice President' của nhân viên có EmployeeID=1
use Northwind
Go
update Employees
set Title = 'Vice President'
where EmployeeID = '1'
Go
```

100 %
Messages
(1 row(s) affected)

12c) Tại kết nối User2, dùng câu lệnh select xem lại kết quả.

TH_Role_Permission_...orthwind (sa (56)) huongDanRole.sql - ...PIT.master (sa (53)) SQLQuery5.sql - LA...thwind (user2 (52))* SQLQuery6.sql - LA...thwind (user3 (54))*

```

select * from Orders
Go
--Kiểm tra quyền select của User2 trên Employees trên CSDL Northwind
select * from Employees
Go

--Câu 4a. xem select để xem thông tin của bảng Employees
select * from Employees
Go
--Câu 4c. User2 dùng câu lệnh select xem lại kết quả.
select * from Employees
Go

```

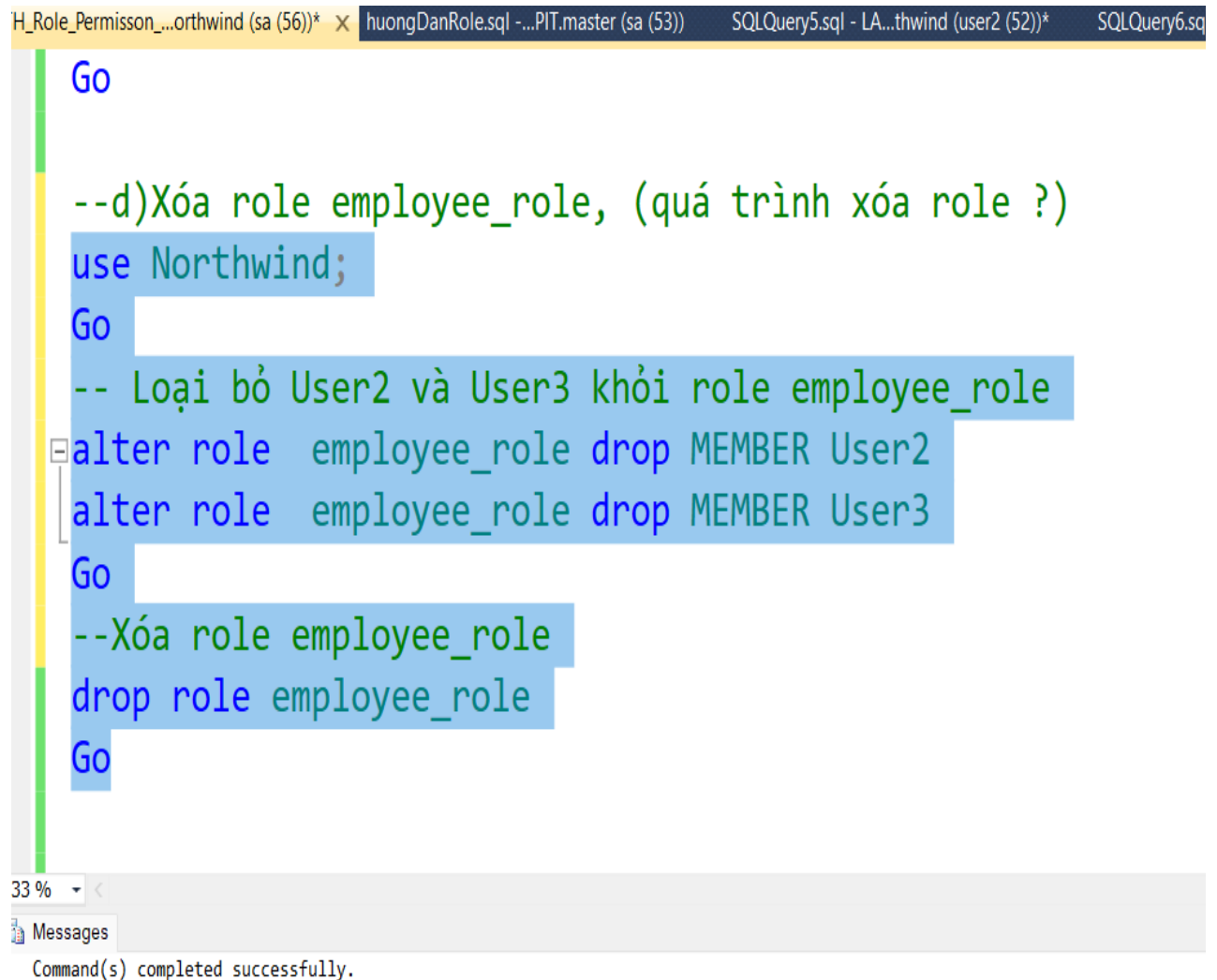
100 % Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate	Address	City	Regi...	PostalCo...	Coun...	HomePhone
1	1	Davolio	Nancy	Vice President	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122	USA	(206) 555-9857
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000	908 W. Capital Way	Tacoma	WA	98401	USA	(206) 555-9482
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000	722 Moss Bay Blvd.	Kirkland	WA	98033	USA	(206) 555-3412
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000	4110 Old Redmond Rd.	Redmond	WA	98052	USA	(206) 555-8122
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17 00:00:00.000	14 Garrett Hill	London	NULL	SW1 8JR	UK	(71) 555-4848
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000	Coventry House Miner Rd.	London	NULL	EC2 7JR	UK	(71) 555-7773
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000	Edgeham Hollow Winchester Way	London	NULL	RG1 9SP	UK	(71) 555-5598
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000	4726 - 11th Ave. N.E.	Seattle	WA	98105	USA	(206) 555-1189
9	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	7 Houndstooth Rd.	London	NULL	WG2 7LT	UK	(71) 555-4444
10	11	Ho	Lam	NULL	NULL	NULL	NULL	NULL	HCM	NULL	NULL	NULL	NULL
11	13	Doe	John	NULL	NULL	NULL	NULL	NULL	USA	NULL	NULL	NULL	NULL

12d) Xóa role employee_role, (quá trình xóa role ?)

-- Loại bỏ User2 và User3 khỏi role employee_role

--Xóa role employee_role



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are several tabs: 'H_Role_Permission_...orthwind (sa (56))*', 'huongDanRole.sql - ...PIT.master (sa (53))', 'SQLQuery5.sql - LA...thwind (user2 (52))*', and 'SQLQuery6.sq'. The active window displays a T-SQL script. The script starts with 'Go' on a new line. Then, it has a comment '--d)Xóa role employee_role, (quá trình xóa role ?)' followed by 'use Northwind;'. Another 'Go' line follows. Then, there is a comment '-- Loại bỏ User2 và User3 khỏi role employee_role'. This is followed by two 'alter role employee_role drop MEMBER' statements, one for 'User2' and one for 'User3'. Another 'Go' line follows. Then, there is a comment '--Xóa role employee_role' followed by 'drop role employee_role'. The script ends with a final 'Go' line. At the bottom, there is a 'Messages' pane showing the message 'Command(s) completed successfully.'

```
Go

--d)Xóa role employee_role, (quá trình xóa role ?)
use Northwind;
Go
-- Loại bỏ User2 và User3 khỏi role employee_role
alter role employee_role drop MEMBER User2
alter role employee_role drop MEMBER User3
Go
--Xóa role employee_role
drop role employee_role
Go
```

33 %

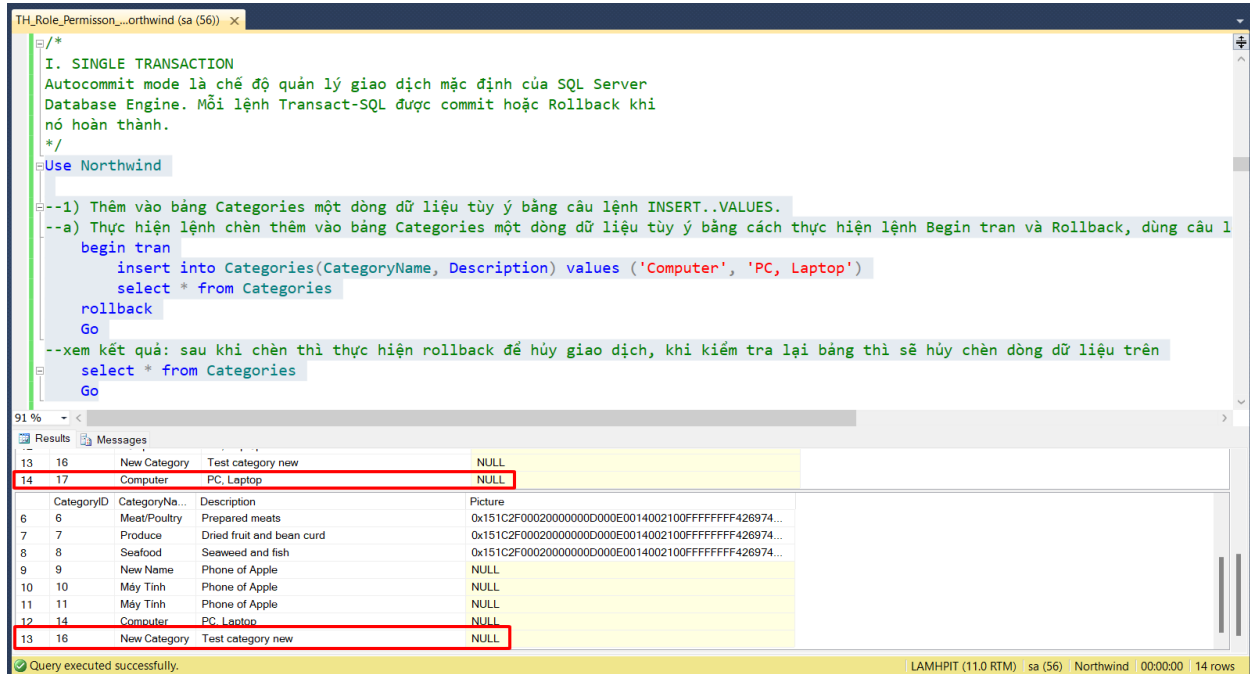
Messages

Command(s) completed successfully.

I.SINGLE TRANSACTION

1) Thêm vào bảng Categories một dòng dữ liệu tùy ý bằng câu lệnh INSERT..VALUES.

1a) Thực hiện lệnh chèn thêm vào bảng Categories một dòng dữ liệu tùy ý bằng cách thực hiện lệnh Begin tran và Rollback, dùng câu lệnh Select * from Categories xem kết quả.



The screenshot displays a SQL Server Enterprise Manager window with a script titled "I. SINGLE TRANSACTION". The script includes a comment about Autocommit mode, followed by a "Use Northwind" statement. It then contains two main steps: 1) Inserting a new category into the Categories table using INSERT..VALUES, and 2) Rolling back the transaction. A final SELECT statement is used to verify the state of the Categories table.

```
/*
I. SINGLE TRANSACTION
Autocommit mode là chế độ quản lý giao dịch mặc định của SQL Server
Database Engine. Mỗi lệnh Transact-SQL được commit hoặc Rollback khi
nó hoàn thành.
*/
Use Northwind

--1) Thêm vào bảng Categories một dòng dữ liệu tùy ý bằng câu lệnh INSERT..VALUES.
--a) Thực hiện lệnh chèn thêm vào bảng Categories một dòng dữ liệu tùy ý bằng cách thực hiện lệnh Begin tran và Rollback, dùng câu l
begin tran
insert into Categories(CategoryName, Description) values ('Computer', 'PC, Laptop')
select * from Categories
rollback
Go

--xem kết quả: sau khi chèn thì thực hiện rollback để hủy giao dịch, khi kiểm tra lại bảng thì sẽ hủy chèn dòng dữ liệu trên
select * from Categories
Go
```

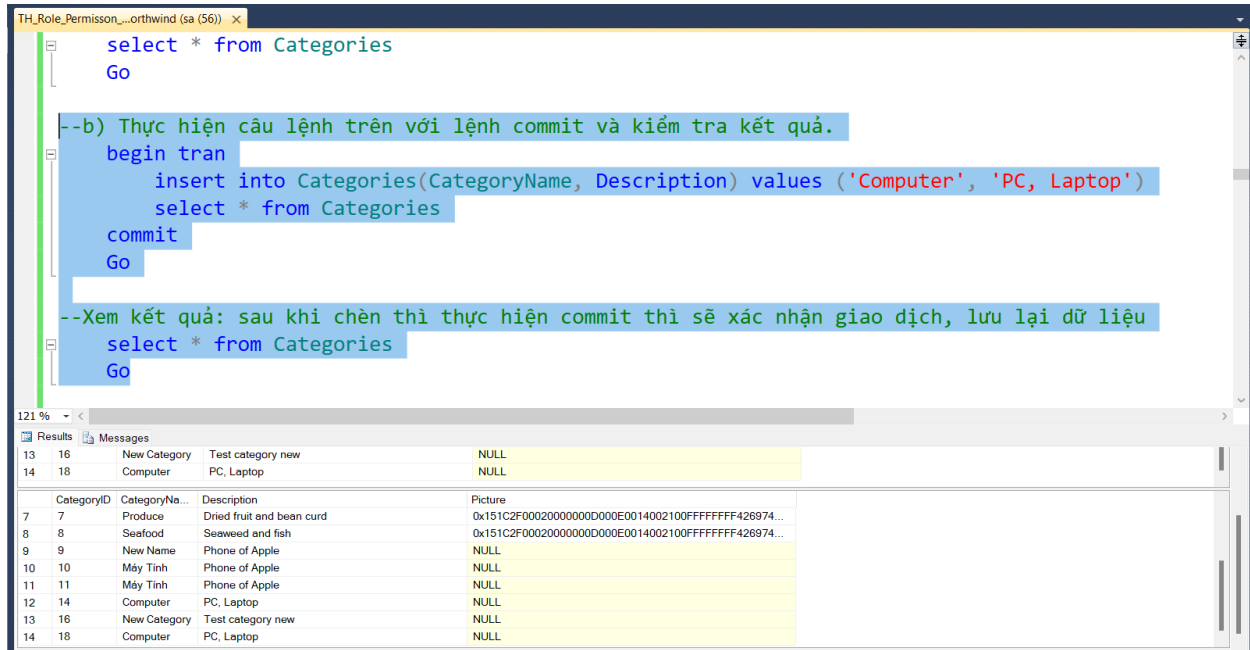
The Results pane shows the output of the SELECT statement. It contains a table with 4 columns: CategoryID, CategoryName, Description, and Picture. The table lists various categories from the Northwind database, including 'New Category', 'Computer', 'Meat/Poultry', 'Produce', 'Seafood', 'New Name', 'Máy Tính', and 'PC, Laptop'. The 'Computer' and 'PC, Laptop' entries are highlighted with red boxes, indicating they were successfully inserted before the rollback.

CategoryID	CategoryName	Description	Picture
13	New Category	Test category new	NULL
14	Computer	PC, Laptop	NULL
6	Meat/Poultry	Prepared meats	0x151C2F0002000000D000E0014002100FFFFFFF426974...
7	Produce	Dried fruit and bean curd	0x151C2F0002000000D000E0014002100FFFFFFF426974...
8	Seafood	Seaweed and fish	0x151C2F0002000000D000E0014002100FFFFFFF426974...
9	New Name	Phone of Apple	NULL
10	Máy Tính	Phone of Apple	NULL
11	Máy Tính	Phone of Apple	NULL
12	Computer	PC, Laptop	NULL
13	New Category	Test category new	NULL

Query executed successfully. LAMHPIT (11.0 RTM) sa (56) Northwind 00:00:00 14 rows

--xem kết quả: sau khi chèn thì thực hiện rollback để hủy giao dịch, khi kiểm tra lại bảng thì sẽ hủy chèn dòng dữ liệu trên

1b) Thực hiện câu lệnh trên với lệnh commit và kiểm tra kết quả.



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query window with the following SQL code:

```
select * from Categories
Go

--b) Thực hiện câu lệnh trên với lệnh commit và kiểm tra kết quả.
begin tran
    insert into Categories(CategoryName, Description) values ('Computer', 'PC, Laptop')
    select * from Categories
commit
Go

--Xem kết quả: sau khi chèn thì thực hiện commit thì sẽ xác nhận giao dịch, lưu lại dữ liệu
select * from Categories
Go
```

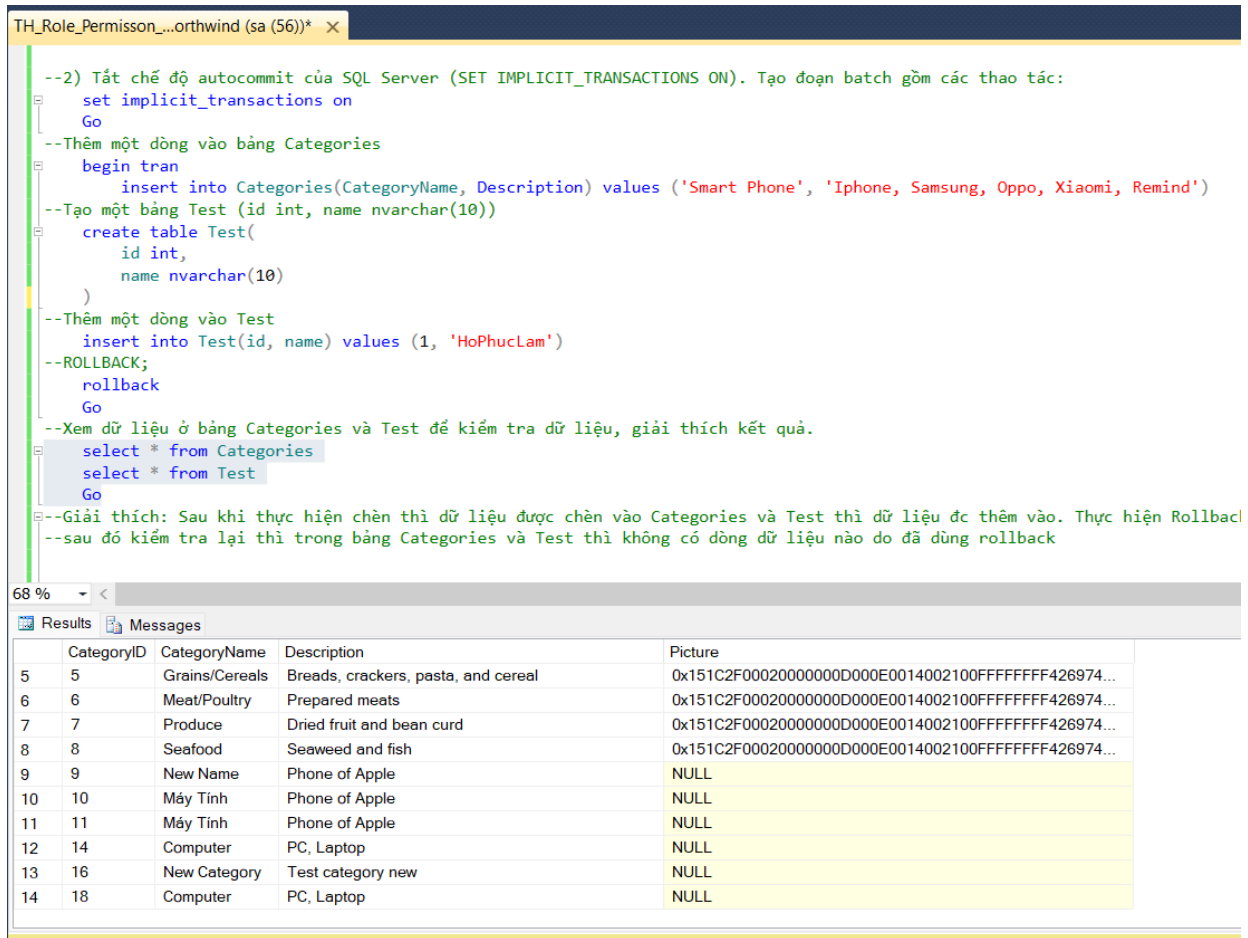
The bottom pane shows the results of the query, displaying a table with 5 columns: CategoryID, CategoryName, Description, Picture, and a fifth column (likely a timestamp or ID). The results are as follows:

CategoryID	CategoryName	Description	Picture	
7	Produce	Dried fruit and bean curd	0x151C2F0002000000D000E0014002100FFFFFFF426974...	
8	Seafood	Seaweed and fish	0x151C2F0002000000D000E0014002100FFFFFFF426974...	
9	New Name	Phone of Apple	NULL	
10	Máy Tính	Phone of Apple	NULL	
11	Máy Tính	Phone of Apple	NULL	
12	Computer	PC, Laptop	NULL	
13	New Category	Test category new	NULL	
14	Computer	PC, Laptop	NULL	

--Xem kết quả: sau khi chèn thì thực hiện commit thì sẽ xác nhận giao dịch, lưu lại dữ liệu

2) Tắt chế độ autocommit của SQL Server (SET IMPLICIT_TRANSACTIONS ON).

Tạo đoạn batch gồm các thao tác:



```
--2) Tắt chế độ autocommit của SQL Server (SET IMPLICIT_TRANSACTIONS ON). Tạo đoạn batch gồm các thao tác:
set implicit_transactions on
Go
--Thêm một dòng vào bảng Categories
begin tran
insert into Categories(CategoryName, Description) values ('Smart Phone', 'Iphone, Samsung, Oppo, Xiaomi, Remind')
--Tạo một bảng Test (id int, name nvarchar(10))
create table Test(
id int,
name nvarchar(10)
)
--Thêm một dòng vào Test
insert into Test(id, name) values (1, 'HoPhucLam')
--ROLLBACK;
rollback
Go
--Xem dữ liệu ở bảng Categories và Test để kiểm tra dữ liệu, giải thích kết quả.
select * from Categories
select * from Test
Go
--Giải thích: Sau khi thực hiện chèn thì dữ liệu được chèn vào Categories và Test thì dữ liệu đc thêm vào. Thực hiện Rollback
--sau đó kiểm tra lại thì trong bảng Categories và Test thì không có dòng dữ liệu nào do đã dùng rollback
```

68 %

Results Messages

	CategoryID	CategoryName	Description	Picture
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F0002000000D000E0014002100FFFFFFFF426974...
6	6	Meat/Poultry	Prepared meats	0x151C2F0002000000D000E0014002100FFFFFFFF426974...
7	7	Produce	Dried fruit and bean curd	0x151C2F0002000000D000E0014002100FFFFFFFF426974...
8	8	Seafood	Seaweed and fish	0x151C2F0002000000D000E0014002100FFFFFFFF426974...
9	9	New Name	Phone of Apple	NULL
10	10	Máy Tính	Phone of Apple	NULL
11	11	Máy Tính	Phone of Apple	NULL
12	14	Computer	PC, Laptop	NULL
13	16	New Category	Test category new	NULL
14	18	Computer	PC, Laptop	NULL

--Giải thích: Sau khi thực hiện chèn thì dữ liệu được chèn vào Categories và Test thì dữ liệu đc thêm vào. Thực hiện Rollback thì sẽ hủy giao dịch và đưa dữ liệu ban đầu,

--sau đó kiểm tra lại thì trong bảng Categories và Test thì không có dòng dữ liệu nào do đã dùng rollback

3) Viết đoạn batch thực hiện các thao tác sau (lưu ý thực hiện lệnh SET XACT_ABORT ON: nếu câu lệnh T-SQL làm phát sinh lỗi run-time, toàn bộ giao dịch được chấm dứt và Rollback)

```
TH_Role_Permission_...orthwind (sa (52)) * x
set xact_abort on
Go
begin tran
--Câu lệnh SELECT với phép chia 0 :SELECT 1/0 as Dummy
select 1/0 as Dummy

--Cập nhật một dòng trên bảng Categories với id='9' (id này không tồn tại)
update Categories
set CategoryName = 'New Name' where CategoryID = '9';

--Xóa một dòng không tồn tại trên bảng Categories (id='66')
delete from Categories where CategoryID = '66';

--Thêm một dòng bất kỳ vào bảng Categories
insert into Categories (CategoryName, Description) values ('New Category Apha', 'Test category Apha');

commit
Go
```

100 % <

Results Messages

Msg 8134, Level 16, State 1, Line 4
Divide by zero error encountered.

```
--Thêm một dòng bất kỳ vào bảng Categories
insert into Categories (CategoryName, Description) values ('New Category Apha', 'Test category Apha');

commit
Go

--Xem lại bảng
select * from Categories
```

100 % <

Results Messages

	CategoryID	CategoryName	Description	Picture
10	10	Máy Tính	Phone of Apple	NULL
11	11	Máy Tính	Phone of Apple	NULL
12	14	Computer	PC, Laptop	NULL
13	16	New Category	Test category new	NULL
14	18	Computer	PC, Laptop	NULL
15	23	New Category	Test category new	NULL

--Kết quả: với set xact_about on thì mọi lỗi run-time trong giao dịch sẽ làm cho toàn bộ giao dịch bị rollback hủy đi.

--Nếu 1/0 đổi thành 1/1 thì giao dịch sẽ không lỗi và thực hiện được tới commit.

4) Thực hiện lệnh SET XACT_ABORT OFF (những câu lệnh lỗi sẽ rollback, transaction vẫn tiếp tục) sau đó thực thi lại các thao tác của đoạn batch ở câu 3. Quan sát kết quả và giải thích kết quả?

```

set xact_abort off
Go
begin tran
    --Câu lệnh SELECT với phép chia 0 để gây lỗi
    select 1/0 as Dummy

    --Cập nhật một dòng trên bảng Categories với id='9' (id này không tồn tại)
    update Categories
    set CategoryName = 'New Name' where CategoryID = '9';

    --Xóa một dòng không tồn tại trên bảng Categories (id='66')
    delete from Categories where CategoryID = '66';

    --Thêm một dòng bất kỳ vào bảng Categories
    insert into Categories (CategoryName, Description) values ('NewApha', 'Test category Apha');

commit

```

100 %

Results Messages

Msg 8134, Level 16, State 1, Line 3
Divide by zero error encountered.

(1 row(s) affected) **update**
(0 row(s) affected) **delete**
(1 row(s) affected) **insert**

```

--Xem lại bảng
select * from Categories

--Kết quả: lỗi khi chia cho 0 sẽ không khiến toàn bộ giao dịch bị rollback chỉ hủy câu lệnh đó.
--Câu lệnh update không gây lỗi, câu lệnh insert sẽ được thêm vào bảng Categories. Commit sẽ thực hiện xác nhận giao dịch.
--Set xact_abort off thì các lỗi run-time sẽ không hủy bỏ toàn bộ giao dịch chỉ ảnh hưởng đến câu lệnh gây lỗi.
--còn các câu lệnh khác trong giao dịch vẫn tiếp tục thực thi

```

83 %

Results Messages

CategoryID	CategoryName	Description	Picture	
12	14	Computer	PC, Laptop	NULL
13	16	New Category	Test category new	NULL
14	18	Computer	PC, Laptop	NULL
15	23	New Category	Test category new	NULL
16	26	NewApha	Test category Apha	NULL
17	27	NewApha	Test category Apha	NULL

--Kết quả: lỗi khi chia cho 0 sẽ không khiến toàn bộ giao dịch bị rollback chỉ hủy câu lệnh đó.

--Câu lệnh update không gây lỗi, câu lệnh insert sẽ được thêm vào bảng Categories. Commit sẽ thực hiện xác nhận giao dịch.

--Set xact_abort off thì các lỗi run-time sẽ không hủy bỏ toàn bộ giao dịch chỉ ảnh hưởng đến câu lệnh gây lỗi. còn các câu lệnh khác trong giao dịch vẫn tiếp tục thực thi

II.CONCURRENT TRANSACTIONS

1)Tạo bảng Accounts (AccountID int NOT NULL PRIMARY KEY,balance int NOT NULL CONSTRAINT unloanable_account CHECK (balance >= 0) Và chèn dữ liệu:

```
--Tạo bảng
create table Accounts(
    AccountID int NOT NULL PRIMARY KEY,
    balance int NOT NULL CONSTRAINT unloanable_account CHECK(balance >= 0)
)
Go

--Chèn dữ liệu
insert into Accounts (AccountID,balance) values (101,1000)
insert into Accounts (AccountID,balance) values (202,2000)

--xem lại bảng
select * from Accounts
Go

--Xóa dữ liệu chèn
delete from Accounts
Go
```

121 %

Results Messages

	AccountID	balan...
1	101	1000
2	202	2000

2) SET TRANSACTION ISOLATION LEVEL

Thiết lập mức độ cô lập là *READ UNCOMMITTED*

```
/*2) SET TRANSACTION ISOLATION LEVEL*/

/*-- Thiết lập mức độ cô lập là READ UNCOMMITTED:
có thể đọc những dòng đang được hiệu chỉnh bởi
các transaction khác nhưng chưa commit */
set transaction isolation level read uncommitted
begin tran
    select * from Employees
commit
Go
```

121 %

Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate	Address
1	1	Davolio	Nancy	Vice President	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000	507 - 20th Ave. E. Apt. 2A
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000	908 W. Capital Way
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000	722 Moss Bay Blvd.
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000	4110 Old Redmond Rd.
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17 00:00:00.000	14 Garrett Hill
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000	Coventry House Miner Rd.
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000	Edgeham Hollow Winch...
8	8	Callahan	Laura	Inside Sales Coordi...	Ms.	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000	4726 - 11th Ave. N.E.
9	9	Dodswor...	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	7 Houndstooth Rd.
10	11	Ho	Lam	NULL	NULL	NULL	NULL	NULL

Thiết lập mức độ cô lập là *READ COMMITTED*

```
/*-- Thiết lập mức độ cô lập là READ COMMITTED
không thể đọc những dòng đang hiệu chỉnh bởi
những transaction khác mà chưa commit*/
set transaction isolation level read committed
begin tran
    select * from Employees
commit
Go
```

121 %

Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate
1	1	Davolio	Nancy	Vice President	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17 00:00:00.000
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000
9	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000
10	11	Ho	Lam	NULL	NULL	NULL	NULL

Thiết lập mức độ cô lập là *REPEATABLE READ*

```
/*-- Thiết lập mức độ cô lập là REPEATABLE READ
Transaction đọc các dòng dữ liệu và đảm bảo rằng các dòng
này sẽ không thay đổi trong suốt quá trình thực thi của transaction.*/
set transaction isolation level repeatable read
begin tran
    select * from Employees
commit
Go
```

121 %

Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate	Address
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000	Coventry House Miner Rd.
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000	Edgeham Hollow Wincheste
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000	4726 - 11th Ave. N.E.
9	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	7 Houndstooth Rd.
10	11	Ho	Lam	NULL	NULL	NULL	NULL	NULL
11	13	Doe	John	NULL	NULL	NULL	NULL	NULL

Thiết lập mức độ cô lập là *SERIALIZABLE*

```
/*-- Thiết lập mức độ cô lập là SERIALIZABLE:Mức độ cô lập cao nhất,
đảm bảo tính nhất quán tuyệt đối bằng cách chặn các transaction
khác có thể ảnh hưởng đến dữ liệu mà transaction hiện tại đang đọc.*/
set transaction isolation level serializable
begin tran
    select * from Employees
commit
Go
```

21 %

Results Messages

	EmployeeID	LastName	FirstNa...	Title	TitleOfCourt...	BirthDate	HireDate	Address	City
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000	Edgeham Hollow Winch...	London
3	8	Callahan	Laura	Inside Sales Coordi...	Ms.	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000	4726 - 11th Ave. N.E.	Seattle
3	9	Dodsw...	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	7 Houndstooth Rd.	London
10	11	Ho	Lam	NULL	NULL	NULL	NULL	NULL	HCM
11	13	Doe	John	NULL	NULL	NULL	NULL	NULL	USA
12	14	Doe	John	NULL	NULL	NULL	NULL	NULL	USA

3) Mở 2 cửa sổ Query của SQL server, thiết lập SET TRANSACTION ISOLATION LEVEL READ COMMITTED ở cả 2 cửa sổ (tạm gọi là clients A bên trái, và client B bên phải).

Client A, client B: cùng thực hiện lệnh SELECT trên bảng Accounts với accountID = 101

```
TH_Role_Permission..._orthwind (sa (52))
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
Quan sát kết quả hiển thị và giải thích.
*/
--Phiên A bên trái:
--thiết lập read committed
set transaction isolation level read committed
begin tran
--xem Accounts
select * from Accounts where AccountID = '101'
--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit;

02082024.sql - LAM..._orthwind (sa (53))
--Câu 3
--Phiên B bên phải:
use Northwind
select * from Accounts
Go
--thiết lập read committed
set transaction isolation level read committed
begin tran
--xem Accounts
select * from Accounts where AccountID = '101'
--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101'
Commit;
```

Clients A cập nhật account trên AccountID =101, balance =1000-200

```
TH_Role_Permission..._orthwind (sa (52))
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
Quan sát kết quả hiển thị và giải thích.
*/
--Phiên A bên trái:
--thiết lập read committed
set transaction isolation level read committed
begin tran
--xem Accounts
select * from Accounts where AccountID = '101'
--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit;

02082024.sql - LAM..._orthwind (sa (53))
--Câu 3
--Phiên B bên phải:
use Northwind
select * from Accounts
Go
--thiết lập read committed
set transaction isolation level read committed
begin tran
--xem Accounts
select * from Accounts where AccountID = '101'
--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101'
Commit;
```

Client B cập nhật account trên AccountID =101, balance =1000-500

```
TH_Role_Permission...orthwind (sa (52)) * X
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
Quan sát kết quả hiển thị và giải thích.
*/
--Phiên A bên trái:
--thiết lập read committed
set transaction isolation level read committed
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit;

02082024.sql - LAMH...(53) Executing... X
--Câu 3
--Phiên B bên phải:
use Northwind
select * from Accounts
Go

--thiết lập read committed
set transaction isolation level read committed
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'

--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit;
```

← Không có message

Client A: SELECT trên Accounts với AccountID =101; COMMIT;

```
TH_Role_Permission...orthwind (sa (52)) * X
--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
--Client B: SELECT trên Accounts với AccountID =101; COMMIT;
Quan sát kết quả hiển thị và giải thích.
*/
--Phiên A bên trái:
--thiết lập read committed
set transaction isolation level read committed
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit;

02082024.sql - LAMH...(53) Executing... X
--Câu 3
--Phiên B bên phải:
use Northwind
select * from Accounts
Go

--thiết lập read commit
set transaction isolati
begin tran

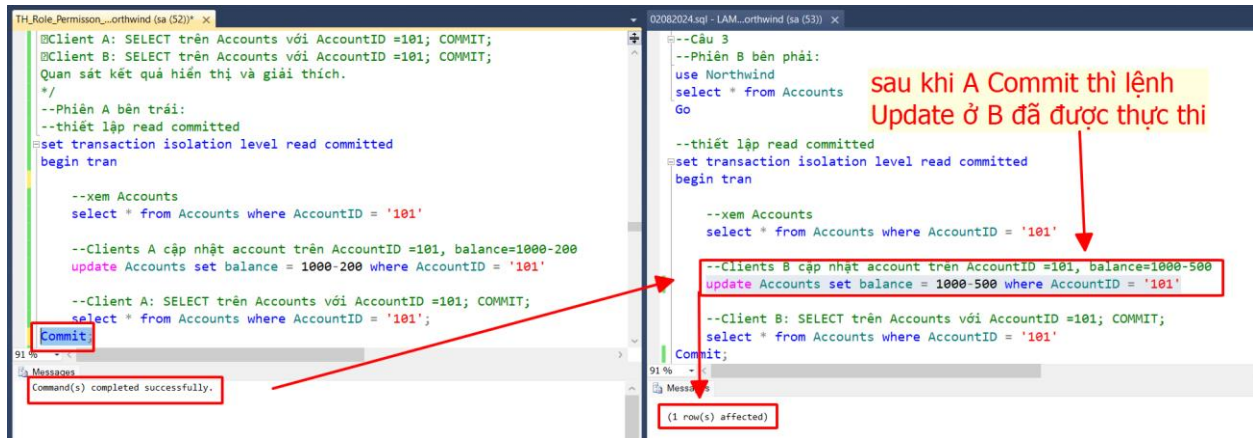
--xem Accounts
select * from Accou

--Clients B cập nhậ
update Accounts set

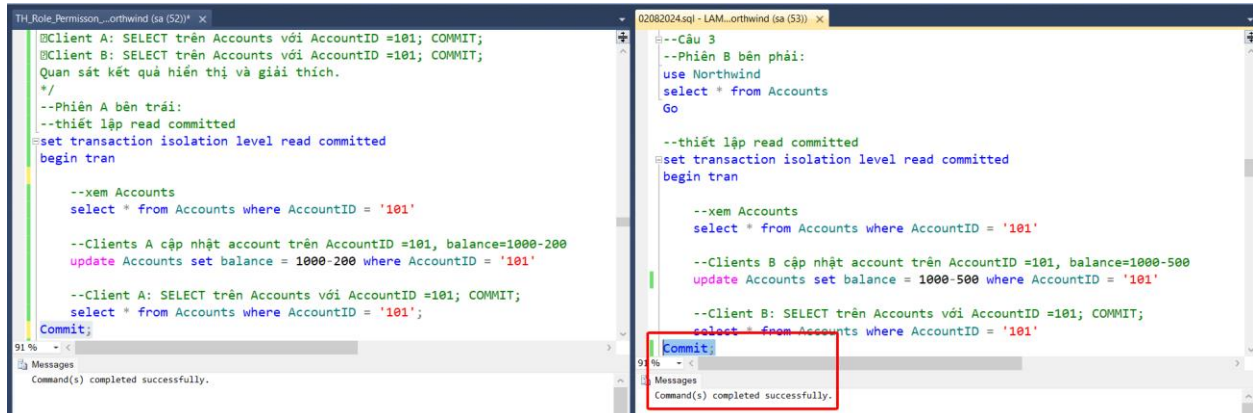
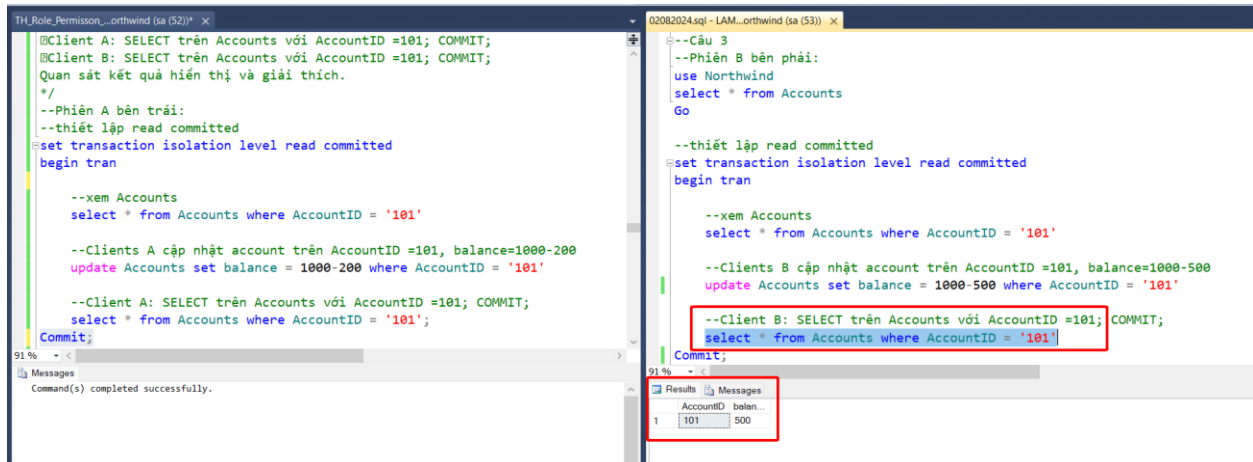
--Client B: SELECT
select * from Accou

Commit;
```

AccountID	balen...
1	101 800



Client B: SELECT trên Accounts với AccountID =101; COMMIT;



Quan sát kết quả hiển thị và giải thích.

Ta thấy sau khi A committ thì lệnh update ở B đã được thực thi do ban đầu đã **SET TRANSACTION ISOLATION LEVEL READ COMMITTED** đảm bảo rằng mỗi transaction chỉ có thể đọc dữ liệu đã được COMMIT bởi các transaction khác. Điều này ngăn chặn hiện tượng "dirty read" (đọc dữ liệu chưa COMMIT từ transaction khác).

4)Thiết lập ISOLATION LEVEL REPEATABLE READ (không thể đọc được dữ liệu đã được hiệu chỉnh nhưng chưa commit bởi các transaction khác và không có transaction khác có thể hiệu chỉnh dữ liệu đã được đọc bởi các giao dịch hiện tại cho đến transaction hiện tại hoàn thành) ở 2 client. Thực hiện yêu cầu sau:

Client A, client B: cùng thực hiện lệnh SELECT trên bảng Accounts với AccountID =101

```
TH_Role_Permission_orthwind (sa (52)) x
/*
--Phiên Client A bên trái
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit

91 %
Results Messages
AccountID balan...
1 101 800
```

```
02082024.sql - LAM_orthwind (sa (53)) x
--Câu 4
--Phiên B bên phải:
--thiết lập repeatable read
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
commit
Go

--Kết quả và giải thích:

91 %
Results Messages
AccountID balan...
1 101 800
```

xem ở bảng ở hai phiên Client A và B

Clients A cập nhật accounts trên AccountID =101, balance=1000-200

```
TH_Role_Permission_T... (52) Executing... X
/*
--Phiên Client A bên trái
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit
```

no message

```
02082024.sql - LAM...orthwind (sa (53)) X
--Câu 4
--Phiên B bên phải:
--thiết lập repeatable read
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
commit
Go

--Kết quả và giải thích:
```

AccountID	balan...
1	101

Client B cập nhật accounts trên AccountID =101, balance =1000-500.

```
TH_Role_Permission...orthwind (sa (52)) X
/*
--Phiên Client A bên trái
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit
```

(1 row(s) affected)

```
02082024.sql - LAM...orthwind (sa (53)) X
--Câu 4
--Phiên B bên phải:
--thiết lập repeatable read
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
commit
Go

--Kết quả và giải thích:
```

Msg 1205, Level 13, State 51, Line 1
Transaction (Process ID 53) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Client A: SELECT trên Accounts với AccountID =101; COMMIT;

```
TH_Role_Permission...orthwind (sa (52)) X
/*
--Phiên Client A bên trái
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients A cập nhật account trên AccountID =101, balance=1000-200
update Accounts set balance = 1000-200 where AccountID = '101'

--Client A: SELECT trên Accounts với AccountID =101; COMMIT;
select * from Accounts where AccountID = '101';
Commit
```

(1 row(s) affected)

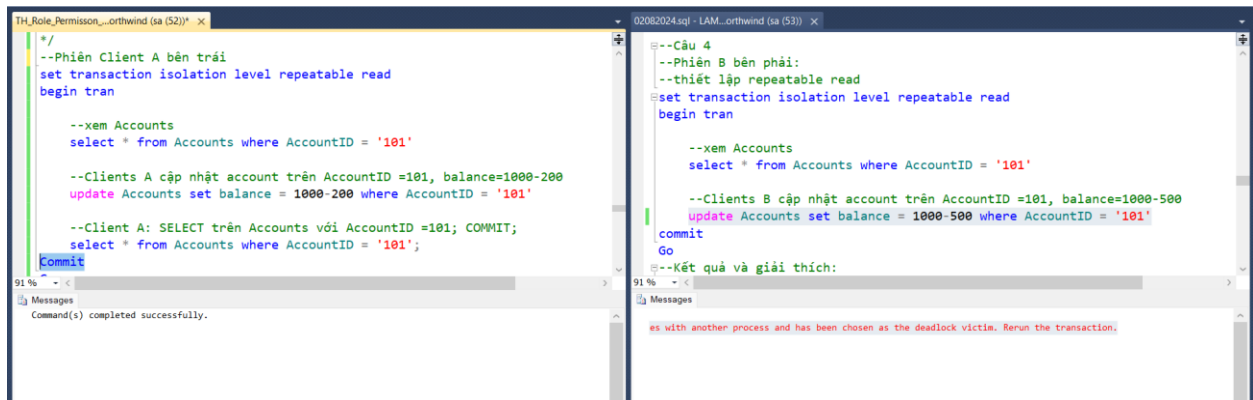
```
02082024.sql - LAM...orthwind (sa (53)) X
--Câu 4
--Phiên B bên phải:
--thiết lập repeatable read
set transaction isolation level repeatable read
begin tran

--xem Accounts
select * from Accounts where AccountID = '101'

--Clients B cập nhật account trên AccountID =101, balance=1000-500
update Accounts set balance = 1000-500 where AccountID = '101'
commit
Go

--Kết quả và giải thích:
```

es with another process and has been chosen as the deadlock victim. Rerun the transaction.

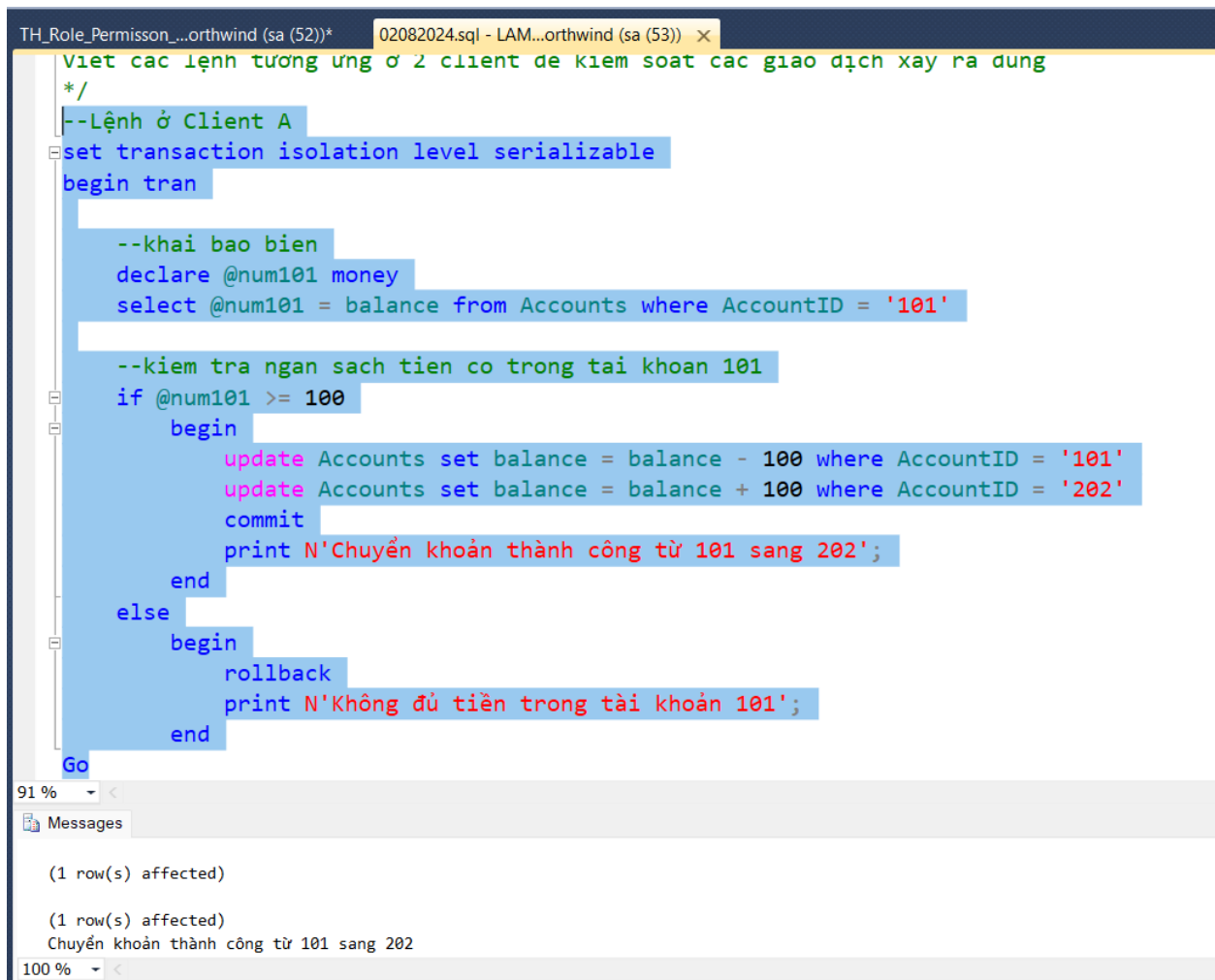


Quan sát kết quả hiển thị và giải thích.

- Mức độ cô lập REPEATABLE READ đảm bảo rằng dữ liệu mà một transaction đọc sẽ không bị thay đổi bởi các transaction khác cho đến khi transaction hiện tại hoàn thành. Điều này làm cho Client A và Client B không thể nhìn thấy những thay đổi của nhau cho đến khi mỗi transaction hoàn thành (COMMIT).
- Client A: Sau khi thực hiện COMMIT, dữ liệu mà Client A thấy sẽ là dữ liệu mà nó đã đọc ở lần SELECT đầu tiên, trước khi Client B thực hiện bất kỳ thay đổi nào. Do đó, nếu không có xung đột khóa, Client A sẽ thấy balance của AccountID = 101 là 800 (vì nó đã cập nhật giảm từ 1000 xuống 800).
- Client B: Sau khi thực hiện UPDATE, balance của AccountID = 101 sẽ là 500. Tuy nhiên, vì Client A đã đọc dữ liệu trước khi Client B cập nhật, nên khi Client A SELECT lại sau khi Client B cập nhật, nó vẫn sẽ thấy balance là 800 cho đến khi COMMIT.

5) Giả sử có 2 giao dịch chuyển tiền từ tài khoản 101 và 202 như sau:

Client A chuyển 100\$ từ tài khoản 101 sang 202



The screenshot shows a SQL Server Enterprise Manager window with a T-SQL script editor. The script is written in Vietnamese and implements a money transfer transaction. It starts with a comment in Vietnamese, followed by a batch separator. The script then sets the transaction isolation level to serializable and begins a transaction. It declares a variable @num101 of type money and selects the balance from the Accounts table where AccountID is '101'. Next, it checks if @num101 is greater than or equal to 100. If true, it updates the balance of AccountID '101' by subtracting 100 and the balance of AccountID '202' by adding 100, then commits the transaction and prints a success message. If false, it rolls back the transaction and prints an error message. The script ends with a 'Go' button. Below the script editor, the Messages pane shows the execution results: two rows affected and a success message.

```
TH_Role_Permission_...orthwind (sa (52))* 02082024.sql - LAM...orthwind (sa (53)) x
Viết các lệnh tương ứng ở 2 client để kiểm soát các giao dịch xảy ra đúng
*/
--Lệnh ở Client A
set transaction isolation level serializable
begin tran

--khai báo biến
declare @num101 money
select @num101 = balance from Accounts where AccountID = '101'

--kiểm tra ngân sách tiền có trong tài khoản 101
if @num101 >= 100
begin
    update Accounts set balance = balance - 100 where AccountID = '101'
    update Accounts set balance = balance + 100 where AccountID = '202'
    commit
    print N'Chuyển khoản thành công từ 101 sang 202';
end
else
begin
    rollback
    print N'Không đủ tiền trong tài khoản 101';
end
Go
```

91 % <

Messages

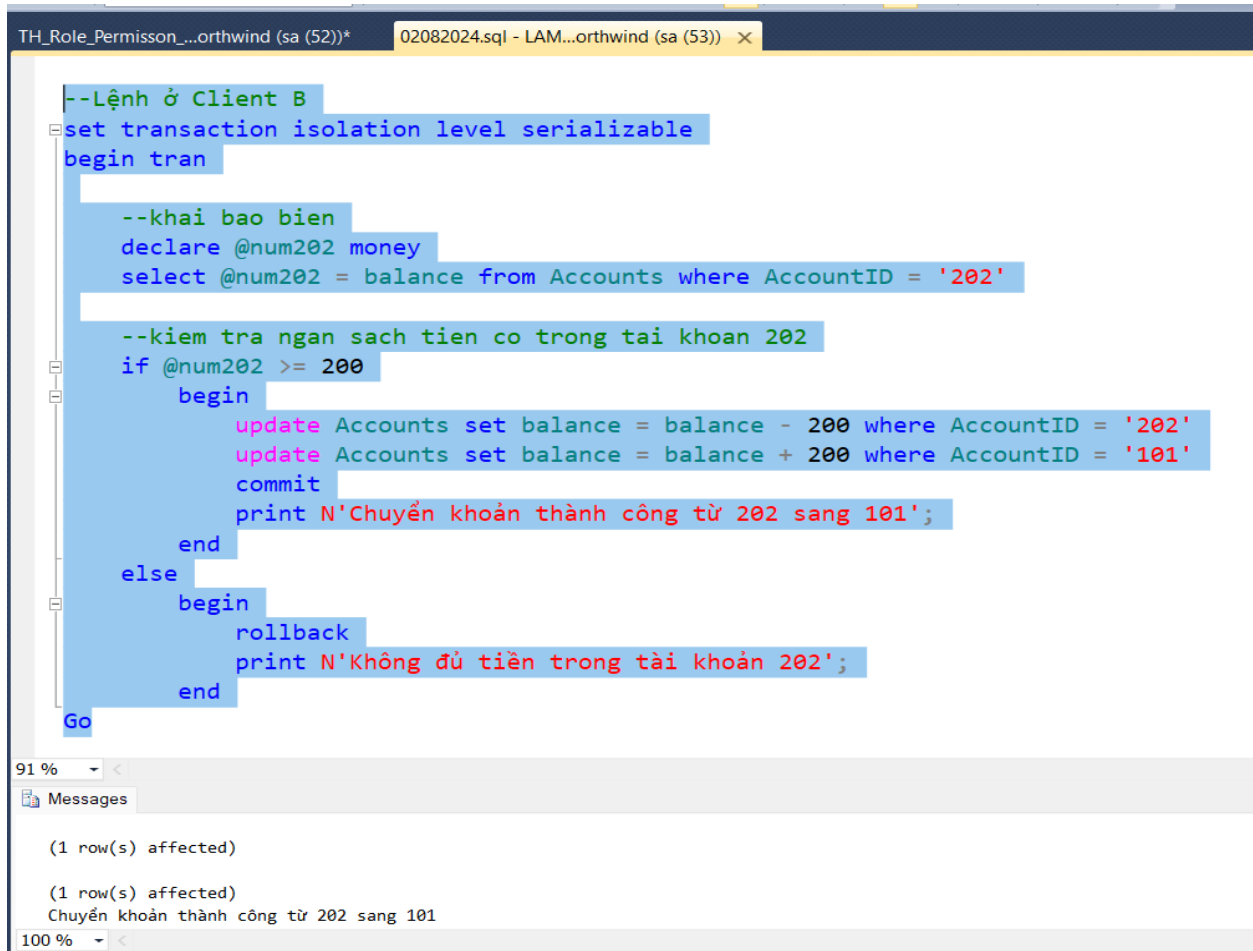
(1 row(s) affected)

(1 row(s) affected)

Chuyển khoản thành công từ 101 sang 202

100 % <

Client B chuyển 200\$ từ tài khoản 202 sang 101.



```
--Lệnh ở Client B
set transaction isolation level serializable
begin tran

--khai bao bien
declare @num202 money
select @num202 = balance from Accounts where AccountID = '202'

--kiem tra ngan sach tien co trong tai khoan 202
if @num202 >= 200
begin
    update Accounts set balance = balance - 200 where AccountID = '202'
    update Accounts set balance = balance + 200 where AccountID = '101'
    commit
    print N'Chuyển khoản thành công từ 202 sang 101';
end
else
begin
    rollback
    print N'Không đủ tiền trong tài khoản 202';
end
end
Go
```

91 %

Messages

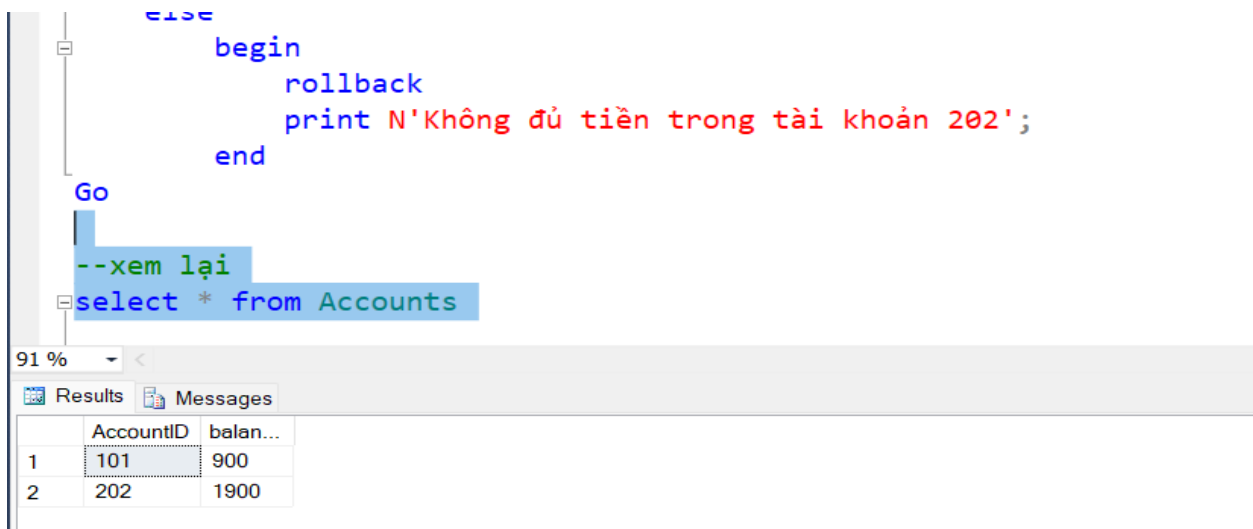
(1 row(s) affected)

(1 row(s) affected)

Chuyển khoản thành công từ 202 sang 101

100 %

--Xem lại



```
end
begin
    rollback
    print N'Không đủ tiền trong tài khoản 202';
end
end
Go

--xem lại
select * from Accounts
```

91 %

Results Messages

	AccountID	balan...
1	101	900
2	202	1900

6)Xóa tất cả dữ liệu của bảng Accounts. Thêm lại các dòng mới INSERT INTO Accounts (AccountID ,balance) VALUES (101,1000); INSERT INTO Accounts (AccountID ,balance) VALUES (202,2000);

```
6)Xóa tất cả dữ liệu của bảng Accounts. Thêm lại các dòng mới
INSERT INTO Accounts (AccountID ,balance) VALUES (101,1000);
INSERT INTO Accounts (AccountID ,balance) VALUES (202,2000);
Quan sát kết quả và giải thích.
*/
--Xóa dữ liệu chèn
delete from Accounts
Go

--Chèn dữ liệu
insert into Accounts (AccountID,balance) values (101,1000)
insert into Accounts (AccountID,balance) values (202,2000)

--xem lại bảng
select * from Accounts
```

91 %

Results Messages

	AccountID	balan...
1	101	1000
2	202	2000

Client A: cập nhật balance của account giảm đi 100 cho **AccountID** =101, cập nhật balance của account tăng lên 100 cho **AccountID** =202

```
--Client A bên trái
/*Client A: cập nhật balance của account giảm đi 100 cho
AccountID =101, cập nhật balance của account tăng lên 100 cho
AccountID =202*/
begin tran
    update Accounts set balance = balance - 100 where AccountID = '101'
    update Accounts set balance = balance + 100 where AccountID = '202'
--Client A thực hiện lệnh
rollback;
select * from Accounts
commit;
```

91 %

Messages

(1 row(s) affected)

(1 row(s) affected)

Client B: thiết lập **ISOLATION LEVEL READ UNCOMMITTED** **SELECT * FROM Accounts; COMMIT;**

```
--Câu 6:
--Client B: thiết lập ISOLATION LEVEL READ UNCOMMITTED
set transaction isolation level read uncommitted
begin tran
    select * from Accounts
commit
Go
```

91 %

Results Messages

	AccountID	balan...
1	101	900
2	202	2100

Client A: ROLLBACK; SELECT * FROM Accounts; COMMIT;

The image shows two side-by-side screenshots of SQL Server Enterprise Manager. The left window, titled 'T1_Role_Permission..._orthwind (sa (52))', contains a script for Client A. It starts with an insert into the Accounts table, followed by a select statement to view the table. Then, it shows a transaction where Client A updates the balance of AccountID 101 (decreasing by 100) and AccountID 202 (increasing by 100). After the transaction, Client A executes a rollback, followed by a select * from Accounts and a commit. The right window, titled '02082024.sql - IAM..._orthwind (sa (53))', contains a script for Client B. It starts with a comment about Client A's updates, followed by a transaction where Client B updates the balance of AccountID 101 (decreasing by 100) and AccountID 202 (increasing by 100). Then, Client B executes a rollback, followed by a select * from Accounts and a commit. Below the scripts, the 'Results' tab shows the state of the Accounts table. In the left window, the table has two rows: AccountID 101 with balance 1000, and AccountID 202 with balance 2000. In the right window, the table has two rows: AccountID 101 with balance 800, and AccountID 202 with balance 2200.

```
insert into Accounts (AccountID,balance) values (202,2000)

--xem lại bảng
select * from Accounts
go

--Client A bên trái
/*Client A: cập nhật balance của account giảm đi 100 cho
AccountID =101, cập nhật balance của account tăng lên 100 cho
AccountID =202*/
begin tran
    update Accounts set balance = balance - 100 where AccountID = '101'
    update Accounts set balance = balance + 100 where AccountID = '202'

--client A thực hiện lệnh
rollback;
select * from Accounts
commit;

/*Client A: cập nhật balance của account giảm đi 100 cho
AccountID =101, cập nhật balance của account tăng lên 100 cho
AccountID =202*/
begin tran
    update Accounts set balance = balance - 100 where AccountID = '101'
    update Accounts set balance = balance + 100 where AccountID = '202'

--client A thực hiện lệnh
rollback;
select * from Accounts
commit;

--Câu 6:
--Client B: thiết lập ISOLATION LEVEL READ UNCOMMITTED
set transaction isolation level read uncommitted
begin tran
    select * from Accounts
commit
```

AccountID	balance
101	1000
202	2000

AccountID	balance
101	800
202	2200

Quan sát kết quả và giải thích.

- Với mức độ cô lập READ UNCOMMITTED, Client B có thể đọc dữ liệu mà không phải chờ đợi cho đến khi transaction của Client A hoàn thành. Do Client A đã ROLLBACK, không có thay đổi nào được áp dụng vào bảng Accounts, vì vậy Client B thấy dữ liệu như trước khi Client A thực hiện bất kỳ thay đổi nào.
- Client A thực hiện các thao tác cập nhật nhưng sau đó hủy bỏ (ROLLBACK) các thay đổi, do đó không có thay đổi nào được áp dụng vào bảng Accounts.
- Client B với mức độ cô lập READ UNCOMMITTED có thể đọc dữ liệu mà không bị chặn bởi các transaction khác đặt mức độ cô lập cao hơn. Do Client A đã ROLLBACK, nên dữ liệu mà Client B đọc là dữ liệu ban đầu của bảng Accounts.

7) Xóa tất cả dữ liệu của bảng Account, thêm lại các dòng mới

INSERT INTO Accounts (AccountID ,balance) VALUES (101,1000); INSERT INTO Accounts (AccountID ,balance) VALUES (202,2000);

```
/*7)Xóa tất cả dữ liệu của bảng Account, thêm lại các dòng mới
INSERT INTO Accounts (AccountID ,balance) VALUES (101,1000); INSERT INTO Accounts (AccountID ,balance) VALUES (202,2000);
--Xóa dữ liệu chèn
delete from Accounts
Go

--Chèn dữ liệu
insert into Accounts (AccountID,balance) values (101,1000)
insert into Accounts (AccountID,balance) values (202,2000)

--xem lại bảng
select * from Accounts
Go
```

1 %

Results Messages

	AccountID	balan...
1	101	1000
2	202	2000

Client A: thiết lập ISOLATION LEVEL REPEATABLE READ; Lấy ra các Accounts có Balance > 1000

```
--Phía Phiên Client A
--Client A: thiết lập ISOLATION LEVEL REPEATABLE READ;
--Lấy ra các Accounts có Balance > 1000
set transaction isolation level repeatable read
begin tran
    select * from Accounts where balance > '1000'

--client a: select * FROM Accounts WHERE balance > 1000; COMMIT;
select * from Accounts where balance > '1000'
commit
```

91 %

Results Messages

	AccountID	balan...
1	202	2000

Client B: INSERT INTO Accounts (AccountID, balance) VALUES (303, 3000); COMMIT;

```
--xem lại bảng
select * from Accounts
Go

--Phía Phiên Client A
--Client A: thiết lập ISOLATION LEVEL REPEATABLE READ;
--Lấy ra các Accounts có Balance>1000
set transaction isolation level repeatable read
begin tran
    select * from Accounts where balance > '1000'

--client a: select * FROM Accounts WHERE balance > 1000; COMMIT;
select * from Accounts where balance > '1000'
commit

--client B: thực hiện lệnh
rollback;
select * from Accounts
commit;

--Câu 6:
--Client B: thiết lập ISOLATION LEVEL READ UNCOMMITTED
set transaction isolation level read uncommitted
begin tran
    select * from Accounts
commit

--Câu 7
--Client B: INSERT INTO Accounts (AccountID, balance) VALUES (303, 3000); COMMIT;
insert into Accounts (AccountID, balance) values (303, 3000)
commit

(1 row(s) affected)
```

*Client A: SELECT * FROM Accounts WHERE balance > 1000; COMMIT;*

```
--xem lại bảng
select * from Accounts
Go

--Phía Phiên Client A
--Client A: thiết lập ISOLATION LEVEL REPEATABLE READ;
--Lấy ra các Accounts có Balance>1000
set transaction isolation level repeatable read
begin tran
    select * from Accounts where balance > '1000'

--client a: select * FROM Accounts WHERE balance > 1000; COMMIT;
select * from Accounts where balance > '1000'
commit

--client A thực hiện lệnh
rollback;
select * from Accounts
commit;

--Câu 6:
--Client B: thiết lập ISOLATION LEVEL READ UNCOMMITTED
set transaction isolation level read uncommitted
begin tran
    select * from Accounts
commit

--Câu 7
--Client B: INSERT INTO Accounts (AccountID, balance) VALUES (303, 3000); COMMIT;
insert into Accounts (AccountID, balance) values (303, 3000)
commit

Msg 3902, Level 16, State 1, Line 1
The COMMIT TRANSACTION request has no corresponding BEGIN TRANSACTION
```

	AccountID	balan...
1	202	2000
2	303	3000

Quan sát kết quả và giải thích:

- Sau khi Client B thêm AccountID = 303 với balance = 3000 và COMMIT, dòng này được thêm vào bảng Accounts. Khi Client A thực hiện SELECT WHERE balance > 1000 lần thứ hai, nó sẽ bao gồm cả AccountID = 202 (từ lần SELECT trước đó) và AccountID = 303 (do Client B đã thêm vào). Điều này thể hiện tính nhất quán của dữ liệu khi sử dụng mức độ cô lập REPEATABLE READ.
- Mức độ cô lập REPEATABLE READ đảm bảo rằng dữ liệu mà Client A đọc được sẽ không thay đổi trong suốt quá trình thực thi của transaction, bất kể các transaction khác có thêm, sửa đổi hoặc xóa bất kỳ dữ liệu nào.
- Client A lấy ra các Accounts có Balance > 1000 và chỉ thấy dòng dữ liệu thỏa điều kiện này là AccountID = 202.
- Sau khi Client B thêm dòng mới và COMMIT, Client A thực hiện lại SELECT và thấy được cả AccountID = 202 và AccountID = 303.