

BÀI TẬP TUẦN 8-9

Mục tiêu:

- 👉 Role, Permission
- 👉 Transaction

Sử dụng SSMS để thực hiện các thao tác sau:

- 1) Tạo một Windows user: User1 (mở Control panel > Administrative tools > Computer Management)
- 2) Tạo một login Windows Authentication cho User1, sau đó login vào sql bằng Windows Authentication của User1 (Khởi động máy đăng nhập hệ thống bằng tài khoản của User1)
- 3) Khởi động SQL, login vào sql bằng Windows Authentication. Kiểm tra quyền truy cập cơ sở dữ liệu của User1 (Mở một CSDL bất kỳ, có thực hiện được không)
- 4) Gán quyền cho User1 để có thể thực hiện tất cả các thao tác trên CSDL NorthWind
- 5) Đăng nhập lại vào SQL bằng windows authentication thông qua tài khoản **user1** và thực hiện các thao tác truy cập CSDL. Logout.
- 6) Đăng nhập lại vào sql bằng SQL server authentication, tài khoản sa. Sử dụng T- SQL.
- 7) Tạo hai login SQL server Authentication **User2 và User3**
- 8) Tạo một database user **User2** ứng với login **User2** và một database user **User3** ứng với login User3 trên CSDL NorthWind.
- 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?
- 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
- 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.
- 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:
 - a) Tại kết nối với User2, thực hiện câu lệnh select để xem thông tin của bảng Employees
 - b) Tại kết nối của User3, thực hiện cập nhật Title='Vice President' của nhân viên có EmployeeID=1
 - c) Tại kết nối User2, dùng câu lệnh select xem lại kết quả.
 - d) Xóa role employee_role, (quá trình xóa role ?)

Mục tiêu:

- Sinh viên hiểu khái niệm về transaction, 4 thuộc tính cơ bản của transaction, cơ chế hoạt động của transaction
- Hiện thực được transaction trong một ngữ cảnh cụ thể

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.

- 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 lệnh Select * from Categories xem kết quả.
 - b) Thực hiện câu lệnh trên với lệnh **commit** và kiểm tra kết quả.
- 2) Tắt chế độ **autocommit** của SQL Server (**SET IMPLICIT_TRANSACTIONS ON**). Tạo đoạn **batch** gồm các thao tác:
 - Thêm một dòng vào bảng **Categories**
 - Tạo một bảng **Test** (id int, name nvarchar(10))
 - Thêm một dòng vào **Test**
 - ROLLBACK;
 - Xem dữ liệu ở bảng **Categories** và **Test** để kiểm tra dữ liệu, giải thích kết quả.
- 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)
 - Câu lệnh SELECT với phép chia 0 :SELECT 1/0 as Dummy
 - Cập nhật một dòng trên bảng **Categories** với id='9' (id này không tồn tại)
 - Xóa một dòng không tồn tại trên bảng **Categories** (id='66')
 - Thêm một dòng bất kỳ vào bảng **Categories**
 - COMMIT;Thực thi đoạn batch, quan sát kết quả và các thông báo lỗi và giải thích kết quả.
- 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?

II. CONCURRENT TRANSACTIONS

- 1) Tạo bảng **Accounts** (AccountID int NOT NULL PRIMARY KEY, balance int NOT NULL

CONSTRAINT unloanable_account CHECK
(balance >= 0)

Chèn dữ liệu:

INSERT INTO Accounts (acctID,balance) VALUES (101,1000);

INSERT INTO Accounts (acctID,balance) VALUES (202,2000);

2) SET TRANSACTION ISOLATION LEVEL

SET TRANSACTION ISOLATION LEVEL

{ READ UNCOMMITTED

| READ COMMITTED

| REPEATABLE READ

| SNAPSHOT

| SERIALIZABLE

}[;]

- **READ UNCOMMITTED**: có thể đọc những dòng đang được hiệu chỉnh bởi các transaction khác nhưng chưa commit
- **READ COMMITTED**: không thể đọc những dòng đang hiệu chỉnh bởi những transaction khác mà chưa commit

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
 - **Clients A** cập nhật account trên **AccountID** =101, balance =1000-200
 - **Client B** cập nhật account trên **AccountID** =101, balance =1000-500
 - **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.

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
- **Clients A** cập nhật accounts trên **AccountID** =101, balance =1000-200
- **Client B** cập nhật accounts trên **AccountID** =101, balance =1000-500.
- **Client A**: SELECT trên Accounts với **AccountID** =101; COMMIT;
- Quan sát kết quả hiển thị và giải thích.

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

- Client B chuyển 200\$ từ tài khoản 202 sang 101.
- Viết các lệnh tương ứng ở 2 client để kiểm soát các giao dịch xảy ra đúng

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);
```

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

ROLLBACK;
SELECT * FROM Accounts;
COMMIT;

Quan sát kết quả và giải thích.

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);
```

- Client A: thiết lập **ISOLATION LEVEL REPEATABLE READ;**
Lấy ra các Accounts có Balance>1000
- Client B:
INSERT INTO Accounts (AccountID ,balance) VALUES (303,3000);
COMMIT;
- Client A:
SELECT * FROM Accounts WHERE balance > 1000;
COMMIT;

Quan sát kết quả và giải thích.