

Sử dụng Stored Procedure để truy xuất dữ liệu

- Trước đây khi dùng Query Analyzer chúng ta có thể đặt tên và save các nhóm câu lệnh SQL vào một file dưới dạng script để có thể sử dụng trở lại sau này. Tuy nhiên thay vì save vào text file ta có thể save vào trong SQL Server dưới dạng Stored Procedure.
- Stored Procedure là một nhóm câu lệnh Transact-SQL đã được compiled (biên dịch) và chứa trong SQL Server dưới một tên nào đó và được xử lý như một đơn vị (chứ không phải nhiều câu SQL riêng lẻ).

Ưu điểm:

- Khi thực thi một câu lệnh SQL thì SQL Server phải kiểm tra permission xem user gọi câu lệnh đó có được phép thực hiện câu lệnh hay không đồng thời kiểm tra cú pháp rồi mới tạo ra một execute plan và thực thi.
- Nếu có nhiều câu lệnh như vậy gửi qua network có thể làm giảm đi tốc độ làm việc của server. SQL Server sẽ làm việc hiệu quả hơn nếu dùng stored procedure vì người gọi chỉ gửi một câu lệnh đơn và SQL Server chỉ kiểm tra một lần sau đó tạo ra một execute plan và thực thi.
- Nếu stored procedure được gọi nhiều lần thì execute plan có thể được sử dụng lại nên sẽ làm việc nhanh hơn. Ngoài ra cú pháp của các câu lệnh SQL đã được SQL Server kiểm tra trước khi save nên nó không cần kiểm tra lại khi thực thi.
- Một khi stored procedure được tạo ra nó có thể được sử dụng lại. Điều này sẽ làm cho việc bảo trì (maintainability) dễ dàng hơn do việc tách rời giữa business rules (tức là những logic thể hiện bên trong stored procedure) và database.
- Ví dụ nếu có một sự thay đổi nào đó về mặt logic thì ta chỉ việc thay đổi code bên trong stored procedure mà thôi. Những ứng dụng dùng stored procedure này có thể sẽ không cần phải thay đổi mà vẫn tương thích với business rule mới.
- Cũng giống như các ngôn ngữ lập trình khác stored procedure cho phép ta đưa vào các input parameters (tham số) và trả về các output parameters đồng thời nó cũng có khả năng gọi các stored procedure khác.
- Security : Giả sử chúng ta muốn giới hạn việc truy xuất dữ liệu trực tiếp của một user nào đó vào một số tables, ta có thể viết một stored procedure để truy xuất dữ liệu và chỉ cho phép user đó được sử dụng stored procedure đã viết sẵn mà thôi chứ không thể "đụng" đến các tables đó

một cách trực tiếp. Ngoài ra stored procedure có thể được encrypt (mã hóa) để tăng cường tính bảo mật.

Cách dùng:

- Tạo stored procedure trong sqlServer
- Tạo SqlCommand với giá trị command text là tên Stored procedure
 - ❖ SqlCommand cmd = new SqlCommand("Tên stored",cn)
 - ❖ Gán CommandType:
 - ❖ cmd.CommandType = CommandType.StoredProcedure
 - ❖ Truyền các tham số vào command nếu StoredProcedure có yêu cầu
 - ❖ Thực thi command

Ví dụ 1:

Ví dụ 1: Stored procedure trả về kết quả của lệnh Select

- string procName = "sp_LoadAccType";
- string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
- SqlConnection Cn = new SqlConnection(CnStr);

- SqlDataAdapter da = new SqlDataAdapter(procName, Cn);
- DataTable dt = new DataTable();
- da.Fill(dt);
- return dt;

Ví dụ 2:

Ví dụ 2: Stored procedure trả về kết quả của lệnh Insert/Delete/Update

- `string CnStr = "Server=.;Database=AgriBankDB;UID=sa;PWD=hoangvy";`
- `SqlConnection Cn = new SqlConnection(CnStr);`
- `string procName = "sp_InsertAccTypes";`
- `SqlCommand cmd = new SqlCommand(procName, Cn);`
- `cmd.CommandType = CommandType.StoredProcedure;`
- `cmd.Parameters.Add("@AccTypeName", SqlDbType.NVarChar);`
- `cmd.Parameters.Add("@Note", SqlDbType.NVarChar);`
- `cmd.Parameters["@AccTypeName"].Value = txtAccTypeName.Text;`
- `cmd.Parameters["@Note"].Value = txtNote.Text;`
- `cmd.Connection.Open();`
- `cmd.ExecuteNonQuery();`
- `cmd.Connection.Close();`

Ví dụ 3:

Ví dụ 3: Stored procedure có dùng tham số output

- `CREATE PROC [dbo].[sp_InsertAccTypes]`
- `@AccTypeName nvarchar(50),`
- `@Note nvarchar(255),`
- `@AccID numeric output,`
- `@err nvarchar(255) output`
- `AS`
- `BEGIN TRAN`
- `INSERT INTO AccTypes(AccTypeName,Note)`
`VALUES(@AccTypeName,@Note)`
- `SET @AccID = @@IDENTITY`
- `SET @err = "`
- `COMMIT TRAN`

Gọi stored procedure và có truyền tham số:

```

string CnStr = "Server=.;Database=AgriBankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);
string procName = "sp_InsertAccTypes";
SqlCommand cmd = new SqlCommand(procName, Cn);
cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@AccTypeName", SqlDbType.NVarChar);
cmd.Parameters.Add("@Note", SqlDbType.NVarChar);
cmd.Parameters.Add("@AccID", SqlDbType.Int).Direction = ParameterDirection.Output;

cmd.Parameters["@AccTypeName"].Value = txtAccTypeName.Text;
cmd.Parameters["@Note"].Value = txtNote.Text;
cmd.Parameters["@AccID"].Value = -1

cmd.Connection.Open();
cmd.ExecuteNonQuery();
cmd.Connection.Close();
string err = cmd.Parameters["@err"].Value.ToString();
int AccId = (int)cmd.Parameters["@AccID"].Value;

```

Ví dụ 4:

Ví dụ 4: Stored procedure có giá trị trả về

- Create Proc TinhTong2
- @soA int,
- @soB int
- AS
- BEGIN
- declare @c int
- set @c= @soA + @SoB
- return @c
- END

```
string procName = "TinhTongz";
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);
SqlCommand cmd = new SqlCommand(procName, Cn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.Add("@SoA", SqlDbType.Int);
cmd.Parameters.Add("@SoB", SqlDbType.Int);
cmd.Parameters["@SoA"].Value = 2;
cmd.Parameters["@SoB"].Value = 3;

SqlParameter p = new SqlParameter("@c", SqlDbType.Int);
p.Direction = ParameterDirection.ReturnValue;
cmd.Parameters.Add(p);

Cn.Open();
cmd.ExecuteNonQuery();
Cn.Close();

MessageBox.Show(cmd.Parameters["@c"].Value.ToString());
```