

Một số phương thức tương tác với SQL Server

Trong các hướng sử dụng sau đây đều thực hiện dạng viết câu lệnh (text) được nhúng trong code và thực thi để trả về danh sách hoặc để thêm mới, cập nhật, xóa thông tin với cơ sở dữ liệu.

Để kết nối và thao tác với cơ sở dữ liệu ta sử dụng không gian tên:
System.Data.SqlClient

1. Lấy danh sách từ cơ sở dữ liệu

```
/// <summary>
/// Hàm hiển thị danh sách người dùng
/// </summary>
private void LoadListNguoiDung()
{
    //Khai báo một biến chứa câu lệnh truy vấn với cơ sở dữ liệu
    string strSQL = "Select * from stanfNguoiDung";

    //Khởi tạo một kết nối từ chuỗi kết nối truyền vào
    SqlConnection conn = new SqlConnection(strConn);

    try
    {
        //Mở kết nối
        conn.Open();

        //Khởi tạo một đối tượng thực thi câu lệnh
        SqlCommand comm = new SqlCommand(strSQL, conn);
        comm.CommandType = CommandType.Text;

        //Khởi tạo một adapter, nó là một đối tượng trung gian để đưa dữ liệu từ database
        lên
```

```
//Sử dụng để hiển thị hoặc đẩy vào dataset hoặc datatable

SqlDataAdapter adapter = new SqlDataAdapter(comm);

//Khởi tạo một dataset
DataSet ds = new DataSet();

//Hoặc có thể khởi tạo một datatable
//DataTable dt = new DataTable();

//Sử dụng phương thức Fill để đẩy dữ liệu lấy được từ database lên dataset,
datatable
//Để hiển thị lên lưới dữ liệu
adapter.Fill(ds);

//Sử dụng DataSource hiển thị dữ liệu lên lưới từ dataset, datatable
gridNguoiDung.DataSource = ds.Tables[0];
}
catch (Exception ex)
{
    //conn.Close();
    MessageBox.Show("Có lỗi xảy ra. Chi tiết lỗi: " + ex.Message);
    //throw ex;
}
finally
{
    //Sau khi dùng xong thì đóng kết nối
    conn.Close();
}
}
```

Trong đó:

```
private string strConn = @"Server=DANGBQ_LAPTOP\SQL2005;
Database=QLSinhVien; uid=sa; pwd=Stanford2012";
```

Đây là chuỗi kết nối với SQL Server với hình thức xác thực qua tài khoản (username, password). Chúng ta có thể sử dụng Windows Authentication để kết nối với cơ sở dữ liệu như sau:

```
private string strConn = @"Server=DANGBQ_LAPTOP\SQL2005;  
Database=QLSinhVien; Integrated Security=true";
```

Sử dụng từ khóa **Integrated Security**

2. Thêm mới một bản ghi vào cơ sở dữ liệu

```
/// <summary>  
/// Thêm mới thông tin người dùng vào cơ sở dữ liệu  
/// </summary>  
/// <returns></returns>  
private int ThemMoiNguoiDung(string strUserName, string strPassword)  
{  
    int intUserId = 0;  
  
    string strInsert = "";  
    //Có thể dùng cách trực tiếp như khai báo sau, nhưng không nên dùng theo cách này  
    //vì có thể  
    //dẫn đến lỗ hổng về bảo mật, lỗi Sql Injection  
    //strInsert = "Insert into stanfNguoiDung(UserName, Password) values  
    ('dangbq','123456')";  
  
    //Sử dụng câu lệnh insert với các tham số truyền vào  
    strInsert = "Insert into stanfNguoiDung(UserName, Password) values (@UserName,  
    @Password)";  
  
    //Khai báo một đối tượng kết nối  
    SqlConnection conn = new SqlConnection(Service.ConnectionString);  
  
    try  
    {  
        //Mở kết nối  
        conn.Open();
```

```
//Khai báo một đối tượng thực thi câu lệnh
SqlCommand command = new SqlCommand(strInsert, conn);

//Khai báo các tham số gồm tên tham số, kiểu dữ liệu của tham số
SqlParameter pUserName = new SqlParameter("@UserName",
SqlDbType.NVarChar);
//Gán giá trị cho tham số lấy từ giao diện qua việc gán vào tham số của hàm
pUserName.Value = strUserName;
//Thêm tham số vào đối tượng thực thi câu lệnh
command.Parameters.Add(pUserName);

SqlParameter pPassword = new SqlParameter("@Password",
SqlDbType.NVarChar);
pPassword.Value = strPassword;
command.Parameters.Add(pPassword);

//Thực thi câu lệnh và về số bản ghi được thực hiện
intUserId = command.ExecuteNonQuery();

//Thực hiện xong thì đóng kết nối
conn.Close();
}
catch (Exception ex)
{
    conn.Close();

    throw ex;
}

return intUserId;
}
```

Trong đó: Sử dụng phương thức **ExecuteNonQuery** để thực hiện thêm mới hoặc cập nhật thông tin, xóa một bản ghi có trong cơ sở dữ liệu.

3. Cập nhật thông tin vào cơ sở dữ liệu

```
/// <summary>
/// Thêm mới thông tin người dùng vào cơ sở dữ liệu
/// </summary>
/// <returns></returns>
private int CapNhatNguoiDung(string strUserName, string strPassword)
{
    int intUserId = 0;

    string strUpdate = "";

    //Sử dụng câu lệnh update với các tham số truyền vào
    strUpdate = "Update stanfNguoiDung SET Password=@Password where
UserName=@UserName";

    //Khai báo một đối tượng kết nối
    SqlConnection conn = new SqlConnection(Service.ConnectionString);

    try
    {
        //Mở kết nối
        conn.Open();

        //Khai báo một đối tượng thực thi câu lệnh
        SqlCommand command = new SqlCommand(strUpdate, conn);

        //Khai báo các tham số gồm tên tham số, kiểu dữ liệu của tham số
        SqlParameter pUserName = new SqlParameter("@UserName",
        SqlDbType.NVarChar);
        //Gán giá trị cho tham số lấy từ giao diện qua việc gán vào tham số của hàm
        pUserName.Value = strUserName;
        //Thêm tham số vào đối tượng thực thi câu lệnh
        command.Parameters.Add(pUserName);

        SqlParameter pPassword = new SqlParameter("@Password",
```

```
SqlDbType.NVarChar);  
    pPassword.Value = strPassword;  
    command.Parameters.Add(pPassword);  
  
    //Thực thi câu lệnh và về số bản ghi được thực hiện  
    intUserId = command.ExecuteNonQuery();  
  
    //Thực hiện xong thì đóng kết nối  
    conn.Close();  
}  
catch (Exception ex)  
{  
    conn.Close();  
  
    throw ex;  
}  
  
return intUserId;  
}
```

4. Xóa thông tin từ cơ sở dữ liệu

```
/// <summary>  
/// Xóa thông tin người dùng  
/// </summary>  
/// <param name="userName"></param>  
/// <returns></returns>  
private int XoaNguoiDung(string userName)  
{  
    int intUserId = 0;  
  
    string strDelete = "";  
    //Khai báo câu lệnh thực hiện  
    strDelete = "Delete from stanfNguoiDung where UserName=@UserName";  
  
    //Khai báo một đối tượng kết nối
```

```
SqlConnection conn = new SqlConnection(Service.ConnectionString);

try
{
    //Mở kết nối
    conn.Open();

    //Khai báo một đối tượng thực thi câu lệnh
    SqlCommand command = new SqlCommand(strDelete, conn);

    //Khai báo tham số gồm tên tham số, kiểu dữ liệu
    SqlParameter pUserName = new SqlParameter("@UserName",
    SqlDbType.NVarChar);
    //Gán giá trị cho tham số
    pUserName.Value = userName;
    //Thêm tham số vào câu lệnh thực thi qua đối tượng comm
    command.Parameters.Add(pUserName);

    //Thực thi câu lệnh
    intUserId = command.ExecuteNonQuery();
}
catch (Exception ex)
{
    //Thông báo lỗi, chỉ xem được với màn hình debug
    //Để chạy đưa ra lỗi thông báo thân thiện với người dùng
    //Ta sử dụng MessageBox.Show để đưa ra thông báo lỗi
    throw ex;
}
finally
{
    //Kết thúc công việc hoặc có lỗi xảy ra thì đều phải đóng kết nối
    //Trước khi hoàn tất phương thức xóa
    conn.Close();
}

return intUserId;
}
```

Ở bài học sau chúng ta sẽ thực thi các phương thức này nhưng sẽ sử dụng viết các thủ tục (Store Procedures) để truy vấn, thêm mới, cập nhật và xóa thông tin.