

Chương 3

KẾT NỐI CSDL

- Giúp sinh viên nắm được
 - Cách tạo kết nối đến cơ sở dữ liệu
 - SQL Server
 - Oracle
 - MS Access
 - Cách quản lý trạng thái của kết nối
 - Cấu hình thông tin kết nối
 - Tạo và cấu hình đối tượng kết nối bằng giao diện

- ◎ Kết nối cơ sở dữ liệu
 - SQL Server
 - Access
 - Oracle
- ◎ Kết nối dùng chung (Connection Pooling)
- ◎ Quản lý trạng thái kết nối
- ◎ Quản lý các sự kiện kết nối
 - StateChanged
 - InfoMessage
- ◎ Tạo & cấu hình đối tượng Connection bằng giao diện

- ◎ Đối tượng thuộc lớp này quản lý và điều khiển việc liên lạc giữa cơ sở dữ liệu và chương trình ứng dụng.
- ◎ Lớp SqlConnection chỉ được dùng cho SQL Server.
- ◎ Hầu hết các thuộc tính, phương thức và sự kiện trong lớp này cũng giống như trong các lớp OleDbConnection và OdbcConnection.

- Các thuộc tính

- String **ConnectionString** { get; set; }
- Int **ConnectionTimeout** { get; }
- String **Database** { get; }
- String **DataSource** { get; }
- Int **PacketSize** { get; }
- String **ServerVersion** { get; }
- ConnectionState **State** { get; }
 - Broken, Closed, Connecting, Executing, Fetching hay Open
- String **WorkstationId** { get; }

- Phương thức
 - SqlTransaction BeginTransaction()
 - Void ChangeDatabase()
 - Void Close()
 - Void Open()
 - SqlCommand CreateCommand()

- Sự kiện
 - StateChangeEventHandler StateChange
 - SqlInfoMessageEventHandlerInfoMessage

- Dùng phương thức tạo lập
 - `SqlConnection ()`
 - `SqlConnection (string connectionString)`
- Các thuộc tính của chuỗi kết nối

Từ khóa	Ý nghĩa
<code>Provider</code> <code>Driver</code>	Trình cung cấp hoặc trình điều khiển cách xử lý truy cập dữ liệu.
<code>DataSource</code> <code>Server</code> <code>Hostname</code>	Tên hoặc địa chỉ của máy chủ chứa cơ sở dữ liệu.
<code>Initial Catalog</code> <code>Database</code>	Tên cơ sở dữ liệu hoặc đường dẫn đến file dữ liệu.
<code>User Id</code> <code>UID</code>	Tên tài khoản đăng nhập vào cơ sở dữ liệu.
<code>Password</code> <code>PWD</code>	Mật khẩu tương ứng với tên tài khoản đăng nhập vào cơ sở dữ liệu.

- Cách 1:

```
SqlConnection mySqlConnection = new SqlConnection();  
mySqlConnection.ConnectionString =  
    "server=localhost;database=Northwind;uid=sa;pwd=sa";
```

- Cách 2:

```
string connectionString = "server=localhost;database=Northwind;uid=sa;pwd=sa";  
SqlConnection mySqlConnection = SqlConnection (connectionString);
```

- Cách 3:

```
SqlConnection mySqlConnection = new SqlConnection  
    ("server=localhost;database=Northwind;uid=sa;pwd=sa");
```

- Cách 4:

```
SqlConnection mySqlConnection = new SqlConnection  
    ("server=localhost;database=Northwind;integrated security=SSPI;");
```


- Cú pháp của chuỗi kết nối
 - provider=Microsoft.Jet.OLEDB.4.0;data source=*databaseFile*

- Ví dụ:

```
string connectionString =
```

```
"provider=Microsoft.Jet.OLEDB.4.0;data source=F:\\Program Files\\Microsoft  
Office\\Office\\Samples\\Northwind.mdb";
```

```
OleDbConnection myOleDbConnection =  
new OleDbConnection (connectionString);
```

- Cú pháp của chuỗi kết nối

- `provider=MSDAORA;data source=OracleNetServiceName;user id=username; password=password`

- Ví dụ

```
string connectionString =  
    "provider=MSDAORA;data source=ORCL;user id=SCOTT;password=TIGER";
```

```
OleDbConnection myOleDbConnection =  
    new OleDbConnection (connectionString);
```

- Trước khi thực thi lệnh, phải mở kết nối bằng cách gọi hàm Open của đối tượng Connection
 - `myConnection.Open()`
- Sau khi thực thi lệnh, phải đóng kết nối bằng cách gọi hàm Close của đối tượng Connection
 - `myConnection.Close()`
- Lưu ý:
 - Khi sử dụng phương thức Fill của DataAdapter, không cần mở và đóng kết nối

- Nhận xét
 - Việc mở và đóng kết nối tốn nhiều thời gian
- Giải pháp
 - Dùng tổ hợp kết nối (hay kết nối dùng chung)
- Đặc điểm
 - Kết nối không bị đóng hẳn khi gọi phương thức Close mà chuyển sang trạng thái chờ (not used)
 - Cải thiện thời gian một cách đáng kể khi thiết lập kết nối cơ sở dữ liệu → tăng hiệu suất

- Max Pool Size (mặc định là 100)
- Min Pool Size (mặc định là 0)
- Nếu cố gắng mở thêm một đối tượng SqlConnection mới trong khi tổ hợp đã đầy, yêu cầu mở sẽ phải chờ cho đến khi có một kết nối cũ bị đóng lại.
- Nếu thời gian chờ lớn hơn số giây được quy định trong thuộc tính ConnectionTimeout, một lỗi sẽ phát sinh.
- Ví dụ

```
SqlConnection mySqlConnection = new SqlConnection(  
    "server=localhost;database=Northwind;uid=sa;pwd=sa;" +  
    "max pool size=10;min pool size=5");
```

- Chương trình 3.3: Connection Pooling
- using System;
- using System.Data;
- using System.Data.SqlClient;
-
- class ConnectionPooling
- {
- public static void Main()
- {
- SqlConnection mySqlConnection =
- new SqlConnection(
- "server=localhost;database=Northwind;uid=sa;pwd=sa; max pool size=10; ");
- for (int count = 1; count <= 10; count++)
- {
- Console.WriteLine("count = " + count);
- **DateTime start = DateTime.Now;**
- **mySqlConnection.Open();**
- **TimeSpan timeTaken = DateTime.Now - start;**
- Console.WriteLine("Milliseconds = " + timeTaken.Milliseconds);
- Console.WriteLine("mySqlConnection.State = " + mySqlConnection.State);
- mySqlConnection.Close();
- }
- }
- }
-

- Kết quả sau khi chạy chương trình
 - count = 1
 - **Milliseconds = 101**
 - mySqlConnection.State = Open
 -
 - count = 2
 - **Milliseconds = 0**
 - mySqlConnection.State = Open
 -
 - count = 3
 - **Milliseconds = 0**
 - mySqlConnection.State = Open
 -
 - count = 4
 - Milliseconds = 0
 - mySqlConnection.State = Open
 -
 - count = 5
 - Milliseconds = 0
 - mySqlConnection.State = Open
 -
 - count = 6
 - Milliseconds = 0
 - mySqlConnection.State = Open

- Trạng thái của kết nối cho biết tình trạng của quá trình yêu cầu kết nối tới cơ sở dữ liệu.
- Để biết trạng thái hiện tại của một kết nối, ta dùng thuộc tính ***State*** của đối tượng `SqlConnection`
- Nên kiểm tra trạng thái trước khi gọi hàm `Open` để mở kết nối

CONSTANT NAME	DESCRIPTION
Broken	Kết nối đã bị ngắt vì một lý do nào đó. Điều này có thể xảy ra sau khi bạn mở kết nối. Để khắc phục, ta gọi hàm đóng kết nối và rồi mở lại kết nối đó.
Closed	Kết nối đã bị đóng.
Connecting	Đang thiết lập kết nối tới cơ sở dữ liệu.
Executing	Kết nối đang được dùng trong quá trình thực thi một lệnh truy vấn.
Fetching	Kết nối đang được dùng trong quá trình nhận thông tin trả về từ cơ sở dữ liệu.
Open	Kết nối đang mở.

- Kiểm tra trạng thái kết nối có thể thực hiện như sau:
 - `if (mySqlConnection.State == ConnectionState.Closed)`
 - `{`
 - `mySqlConnection.Open();`
 - `}`

- Sự kiện StateChange
 - Phát sinh khi trạng thái của kết nối bị thay đổi
 - Dùng để quản lý sự thay đổi về trạng thái của đối tượng Connection
 - Đăng ký trình xử lý sự kiện

```
mySqlConnection.StateChange += new StateChangeEventHandler(StateChangeHandler);
```

```
public static void StateChangeHandler(object mySender, StateChangeEventArgs myEvent)  
{  
    Console.WriteLine("mySqlConnection State has changed from " + myEvent.OriginalState  
        + " to " + myEvent.CurrentState );  
}
```

• Chương trình 3.4: Sử dụng sự kiện StateChange

- using System;
- using System.Data;
- using System.Data.SqlClient;
-
- class StateChange
- {
- public static void StateChangeHandler(
- object mySender, StateChangeEventArgs myEvent)
- {
- Console.WriteLine("mySqlConnection State has changed from " + myEvent.OriginalState
- + " to " + myEvent.CurrentState);
- }
- public static void Main()
- {
- SqlConnection mySqlConnection = new
- SqlConnection("server=localhost;database=Northwind;uid=sa;pwd=sa");
- mySqlConnection.StateChange += new
- StateChangeEventHandler(StateChangeHandler);
- Console.WriteLine("Calling mySqlConnection.Open()");
- mySqlConnection.Open();
- Console.WriteLine("Calling mySqlConnection.Close()");
- mySqlConnection.Close();
- }
- }

- **Kết quả sau khi chạy chương trình:**
 - **Calling `mySqlConnection.Open()`**
 - `mySqlConnection` State has changed from Closed to Open
 - **Calling `mySqlConnection.Close()`**
 - `mySqlConnection` State has changed from Open to Closed

- Sự kiện InfoMessage

- phát sinh khi cơ sở dữ liệu trả về một cảnh báo hoặc một thông điệp.
- dùng để quản lý các thông điệp, cảnh báo hoặc lỗi
- Để lấy thông điệp, ta đọc nội dung thuộc tính Errors của tham số SqlInfoMessageEventArgs
- Đăng ký trình xử lý sự kiện

```
mySqlConnection.InfoMessage += new SqlInfoMessageEventHandler(InfoMessageHandler);
```

```
public static void InfoMessageHandler(object mySender, SqlInfoMessageEventArgs myEvent)  
{  
    Console.WriteLine("The following message was produced:\n" + myEvent.Errors[0] );  
}
```

- **Chương trình 3.5: Sử dụng sự kiện InfoMessage**

- class InfoMessage
- {
- public static void InfoMessageHandler(
• object mySender, SqlInfoMessageEventArgs myEvent)
- {
- Console.WriteLine("The following message was produced:\n" + myEvent.Errors[0]);
- }
- public static void Main()
- {
- SqlConnection mySqlConnection = new
- SqlConnection("server=localhost;database=Northwind;uid=sa;pwd=sa");
- mySqlConnection.InfoMessage += new
- SqlInfoMessageEventHandler(InfoMessageHandler);
- mySqlConnection.Open();
- SqlCommand mySqlCommand = mySqlConnection.CreateCommand();
- mySqlCommand.CommandText = "PRINT 'This is the message from the PRINT statement'";
- mySqlCommand.ExecuteNonQuery();
- mySqlCommand.CommandText = "RAISERROR('This is the message from the RAISERROR statement', 10, 1)";
- mySqlCommand.ExecuteNonQuery();
- mySqlConnection.Close();
- }
- }

- **Kết quả chạy chương trình:**
 - The following message was produced:
 - System.Data.SqlClient.SqlError: This is the message from the PRINT statement
 - The following message was produced:
 - System.Data.SqlClient.SqlError: This is the message from the RAISERROR statement

HẾT CHƯƠNG 3