TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa CNTT – Bộ môn CNPM

LẬP TRÌNH NÂNG CAO

Giảng viên: Nguyễn Thị Phương Dung
Email: dungntp@tlu.edu.vn
SĐT: 0946 079 903

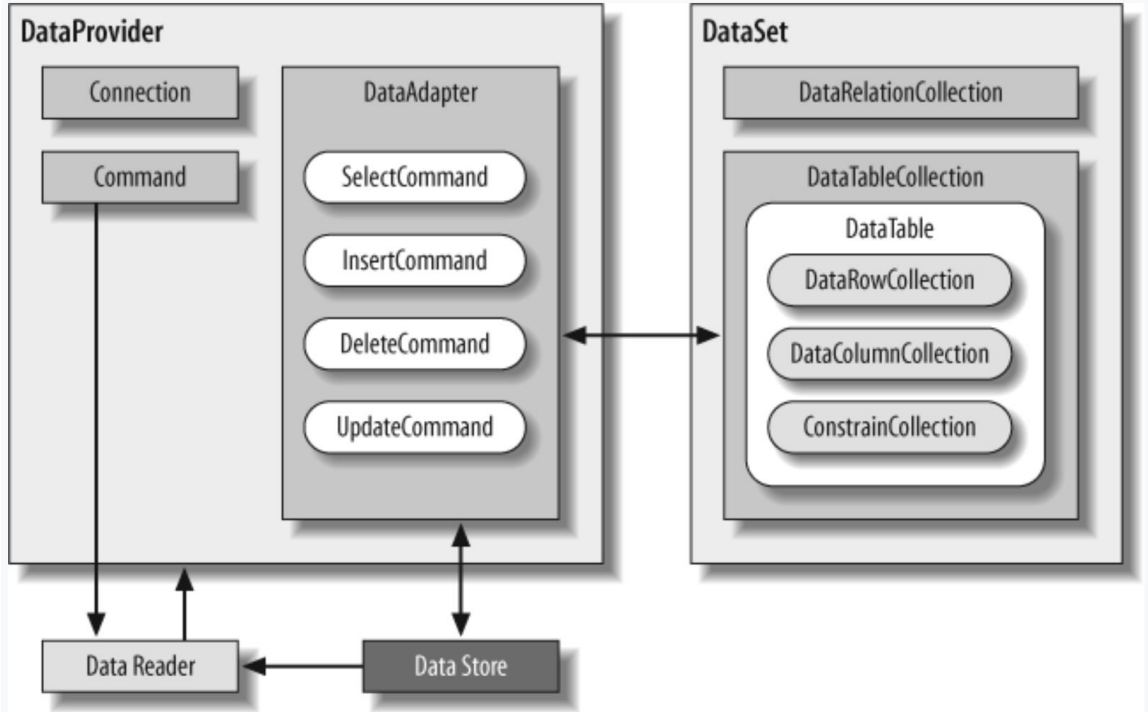
ADO.Net
(ActiveX Data Object.NET)

Giới thiệu về ADO.Net

- Là một bộ thư viện hướng đối tượng (OOP) cho phép tương tác với dữ liệu nguồn.
- Dữ liệu nguồn thường là các cơ sở dữ liệu như: SQLServer, MySQL, Oracle Database,…
- Dữ liệu nguồn cũng có thể là các file text, XML, hoặc excel

Kiến trúc của ADO.Net

- ADO.Net được chia làm 2 phần rời rạc có thể sử dụng độc lập hoặc đồng thời:
  - DataProvider
  - DataSet



DataProvider

- Là các thư viện lớp cung cấp chức năng tạo kết nối đến nguồn dữ liệu, thi hành các lệnh trên nguồn dữ liệu đó.
- Mỗi thư viện hỗ trợ kết nối tới một loại cơ sở dữ liệu khác nhau.

Một số DataProvider phổ biến

DataProvider	API prefix	Loại cơ sở dữ liệu hỗ trợ
ODBC Data Provider	Odbc	Open Database Connectivity
OleDb Data Provider	OleDb	Object Linking and Embedding, Database. VD: Access hoặc Excel.
Oracle Data Provider	Oracle	Oracle Databases.
SQL Data Provider	Sql	Microsoft SQL Server.
Borland Data Provider	Bdp	Sử dụng chung cho nhiều loại cơ sở dữ liệu khác: Interbase, SQL Server, IBM DB2, and Oracle.

Các thành phần của DataProvider

- Bao gồm:
  - Connection: làm nhiệm vụ kết nối tới cơ sở dữ liệu
  - Command: thực hiện các thao tác với CSDL như; select, insert, update, delete
  - DataReader: cung cấp việc đọc từng dòng dữ liệu theo chiều tiến từ đầu đến cuối.
  - DataApdater: đóng vài trò như là cầu nối giữa Dataset và CSDL, tải dữ liệu lên dataset hoặc đồng bộ các thay đổi ở dataset về lại CSDL

DataSet

- Là các thư viện lớp tạo các đối tượng để quản lý dữ liệu không phụ thuộc vào nguồn dữ liệu.
- Có thể được coi là một cơ sở dữ liệu sao chép của cơ sở dữ liệu nguồn.
- Bao gồm nhiều DataTable

Kết nối cơ sở dữ liệu SQL Server với winform

Các đối tượng cần dùng

- SqlConnection: kết nối dữ liệu
- SqlCommand: thực hiện các câu lệnh sql
- SqlDataAdapter: truy vấn dữ liệu
- SqlDataReader: đọc dữ liệu
- DataTable: chứa dữ liệu được truy vấn ra

SqlConnection

- SqlConnection biểu diễn một kết nối liên tục tới nguồn dữ liệu SQL Server
- Trong đó chuỗi kết nối – ConnectionString, xác định server mà đối tượng SqlConnection sẽ sử dụng bao gồm 3 thuộc tính sau:
  - DataSource – xác định tên của server muốn kết nối tới.
  - Initial Catalog – xác định tên của CSDL sử dụng trên server.
  - Integrated Security – chế độ bảo mật.
    - Nếu khi kết nối với SQL server bằng chế độ xác thực của Windows thì Integrated Security = true.
    - Ngược lại, khi kết nối bằng chế độ xác thực của SQL Server thì phải ghi rõ tên đăng nhập và mật khẩu vào 2 thuộc tính UserID và Password (UserID=username;Password=password)

Ví dụ tạo một kết nối SqlConnection

- Ví dụ:

```
string chuoiketnoi = "Data Source=LAPTOP-CJHIRH4S\\SQLEXPRESS; " +
                     " Initial Catalog=QLSV; " +
                     " Integrated Security=True";
SqlConnection conn = null;
private void Form1_Load(object sender, EventArgs e)
{
    conn = new SqlConnection(chuoiketnoi);
    conn.Open();
```

Khai báo biến bên ngoài các hàm để dùng chung trong nhiều hàm (biến toàn cục)

Khởi tạo và Open biến SqlConnection trong sự kiện Load form để đảm bảo form được kết nối CSDL ngay khi chương trình chạy

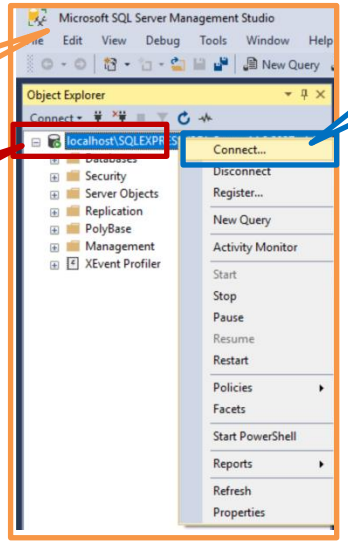
- Để lấy thông tin cho chuỗi kết nối, có thể thực hiện theo 1 trong 2 cách sau:

Cách 1:

- Tự viết chuỗi kết nối bằng tay => phải chú ý viết đúng chính tả
- Để lấy Data Source, thực hiện theo các bước sau:

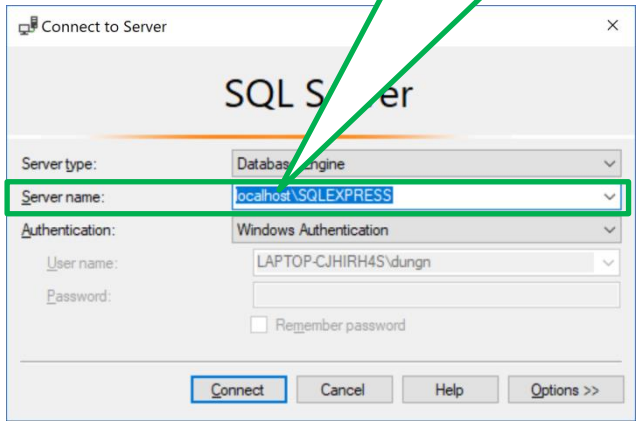
B1: Vào cửa sổ Microsoft SQL Server Management Studio

B2: Bấm chuột phải vào tên server



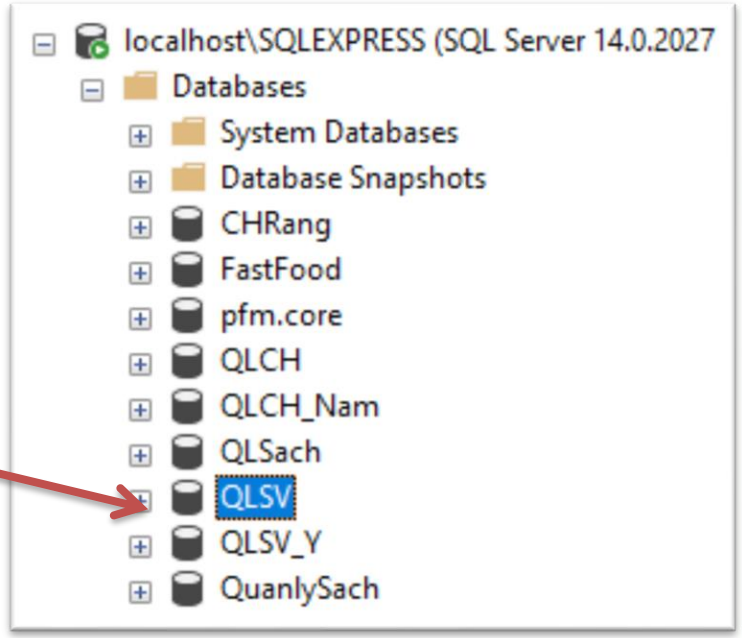
B3: Chọn Connect…

B4: Server name chính là giá trị sẽ gán cho thuộc tính Data Source

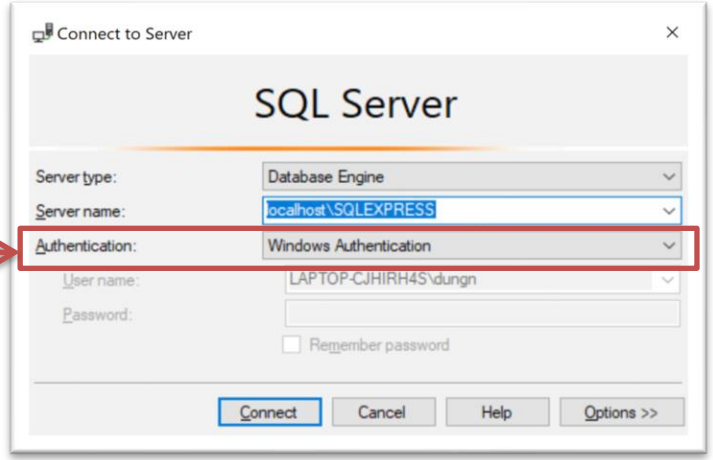


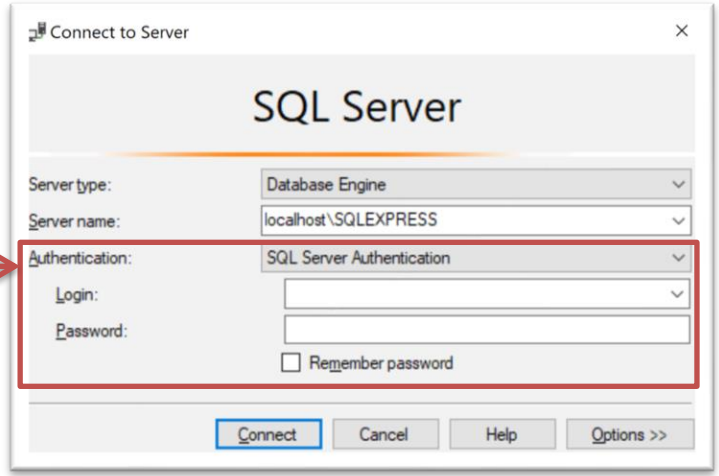
- Initial catalog:

Chính là tên của cơ sở dữ liệu cần kết nối (như trong ví dụ minh họa, Initial catalog = QLSV)



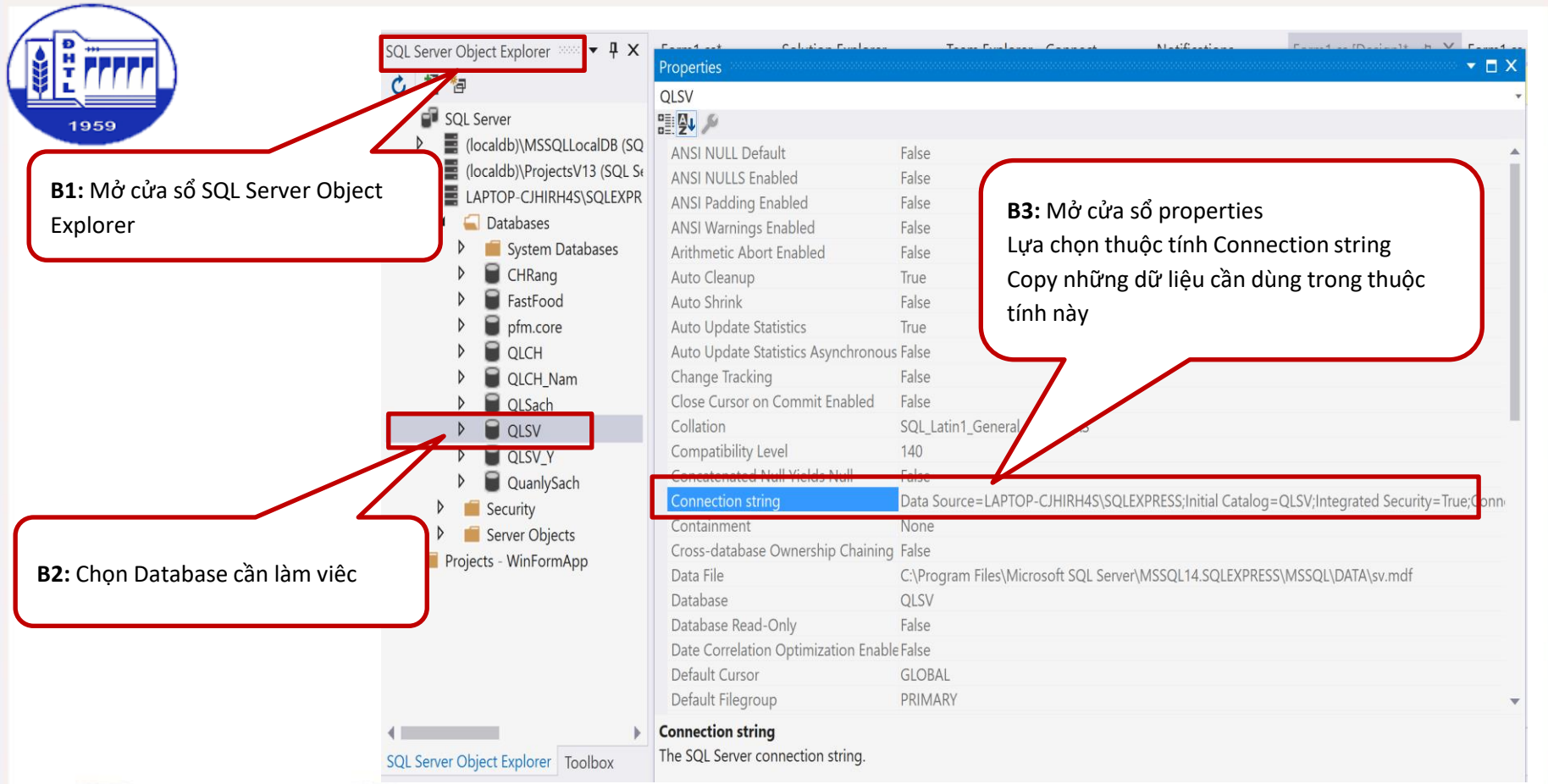
- Integrated Security:
- Thuộc tính này cần chú ý:
  - Nếu Authentication là Windows Authentication thì thiết lập Integrated Security = true.
  - Nếu Authentication là SQL Server Authentication thì phải viết rõ tên đăng nhập và mật khẩu vào 2 thuộc tính UserID và Password (UserID=username;Password =password)





Cách 2

- Thực hiện tạo đối tượng kết nối trong chương trình và copy chuỗi kết nối của đối tượng đó
- Các bước thực hiện:


SQL Server Object Explorer
SQL Server
(localdb)\MSSQLLocalDB (SQ
(localdb)\ProjectsV13 (SQL Se
LAPTOP-CJHIRH4S\SQLEXPR
Databases
System Databases
CHRang
FastFood
pfm.core
QLCH
QLCH_Nam
QLSach
QLSV
QLSV_Y
QuanlySach
Security
Server Objects
Projects - WinFormApp
SQL Server Object Explorer
Toolbox
Properties
QLSV
ANSI NULL Default False
ANSI NULLS Enabled False
ANSI Padding Enabled False
ANSI Warnings Enabled False
Arithmetic Abort Enabled False
Auto Cleanup True
Auto Shrink False
Auto Update Statistics True
Auto Update Statistics Asynchronous False
Change Tracking False
Close Cursor on Commit Enabled False
Collation SQL_Latin1_General
Compatibility Level 140
Connection string Data Source=LAPTOP-CJHIRH4S\SQLEXPRESS;Initial Catalog=QLSV;Integrated Security=True;
Containment None
Cross-database Ownership Chaining False
Data File C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\sv.mdf
Database QLSV
Database Read-Only False
Date Correlation Optimization Enable False
Default Cursor GLOBAL
Default Filegroup PRIMARY
Connection string
The SQL Server connection string.
B1: Mở cửa sổ SQL Server Object Explorer
B2: Chọn Database cần làm việc
B3: Mở cửa sổ properties
Lựa chọn thuộc tính Connection string
Copy những dữ liệu cần dùng trong thuộc tính này

SqlDataAdapter

- Được dùng để truy vấn dữ liệu
- Sử dụng kết hợp với DataTable để chứa dữ liệu
- Ví dụ:

```
private void Form1_Load(object sender, EventArgs e)
{
    conn = new SqlConnection(chuoiketnoi);
    conn.Open();
    string sql = "Select * from SV";
    SqlDataAdapter daSV = new SqlDataAdapter(sql, conn);
    DataTable dtSV = new DataTable();
    daSV.Fill(dtSV);
    dgvSV.DataSource = dtSV;
}
```

Ví dụ sử dụng DataAdapter

```
private void Form1_Load(object sender, Ev
{
    conn = new SqlConnection(chuoiketnoi)
    conn.Open();
    string sql = "Select * from SV";
    SqlDataAdapter daSV = new SqlDataAdapter(sql, conn);
    DataTable dtSV = new DataTable();
    daSV.Fill(dtSV);
    dgvSV.DataSource = dtSV;
}
```

Câu lệnh truy vấn dữ liệu

Khai báo và khởi tạo một biến kiểu SqlDataAdapter với câu lệnh truy vấn và connection đã tạo ở trên

Đổ dữ liệu truy vấn được vào bảng

Hiển thị dữ liệu từ bảng lên giao diện (hiển thị vào DatagridView) thông qua DataTable

Ví dụ sử dụng DataAdapter

- Ví dụ:

```
//lấy dữ liệu vào DataAdapter
SqlDataAdapter daLoaisach = new SqlDataAdapter("SELECT
* FROM LoaiSach", conn);
//đổ dữ liệu vừa lấy ra vào DataTable
DataTable dtLoaiSach = new DataTable();
daLoaiSach.Fill(dtLoaiSach);
//hiển thị dữ liệu lên combobox
cbLoaisach.DataSource = dtLoaiSach;
cbLoaisach.DisplayMember = "TenLoai";
cbLoaisach.ValueMember = "MaLoai";
```

SqlCommand

- Thực hiện một câu lệnh truy vấn trực tiếp tới cơ sở dữ liệu: Select, Insert, Update, Delete hoặc gọi một thủ tục (Store Procedure) từ cơ sở dữ liệu.

Thành phần của SqlCommand

- CommandText: xác định câu lệnh cần thực hiện.
- CommandType: xác định kiểu câu lệnh cần thực hiện.
- Connection: xác định đối tượng kết nối SqlConnection.
- ExecuteNonQuery(): thực hiện các câu lệnh insert, update, delete và trả về số dòng được thực hiện, ngược lại trả về giá trị -1.
- ExecuteReader(): thực hiện câu lệnh select và trả kết quả truy vấn được vào đối tượng SqlDataReader

Ví dụ SqlCommand

- Ví dụ thêm mới dữ liệu vào bảng Sinhviên

```
private void btThem_Click(object sender, EventArgs e)
{
    int gt = 0;
    if(rbNam.Checked == true)
        gt = 1;
    string sql_Insert = "Insert into SV values('" + tbMa.Text + "', N'"
                        + tbHoten.Text + "', '" + dtpNgaysinh.Value.Date
                        + "'," + gt.ToString() + ",'" + cbQue.SelectedValue.ToString() + "')";
    SqlCommand cmd = new SqlCommand(sql_Insert, conn);
    //dùng SqlCommand để thực hiện các câu lệnh cập nhật dữ liệu như Insert, Update, Delete
    cmd.ExecuteNonQuery();//thực hiện câu lệnh trong chuỗi sql_Insert
    //thực hiện cập nhật dữ liệu mới thêm vào datagridview
    dtSV.Rows.Clear();//xóa dữ liệu trong datagridview
    daSV.Fill(dtSV);//hiển thị lại dữ liệu vào datagridview
}
```

DataTable

- DataTable biểu diễn một bảng trong DataSet.
- DataSet có thể chứa nhiều DataTable.
  - ChildRelations: xác định tập hợp các đối tượng DataRelation trỏ tới con của đối tượng DataTable.
  - Clear(): xoá tất cả dữ liệu trong đối tượng DataTable.
  - ColumnChanged(): sự kiện xảy ra khi dữ liệu tại một cột bị thay đổi.
  - ColumnChanging(): sự kiện xảy ra khi dữ liệu tại một cột đang bị thay đổi.
  - Columns: xác định tập hợp các đối tượng DataColumn

DataTable

- Constraints: xác định tập hợp các đối tượng Constraint
- NewRow(): tạo ra một hàng mới, rỗng trong DataTable
- ParentRelations: xác định tập hợp các đối tượng DataRelation trỏ tới cha của đối tượng DataTable.
- PrimaryKey: xác định một mảng các đối tượng DataColumn, cung cấp khoá chính cho đối tượng DataTable.
- RowChanged(): sự kiện xảy ra khi dữ liệu trong đối tượng DataRow bị thay đổi.
- RowChanging(): sự kiện xảy ra khi dữ liệu trong đối tượng DataRow đang bị thay đổi.

DataTable

- RowDeleted(): sự kiện xảy ra khi một hàng bị xoá.
- RowDeleting(): sự kiện xảy ra khi một hàng đang bị xoá.
- Rows: xác định tập hợp các đối tượng DataRow.
- Select(): xác định mảng các đối tượng DataRow thoả mãn tiêu chuẩn lựa chọn.
- TableName: xác định tên của đối tượng DataTable.

Ví dụ

- Sử dụng SQLCommand để thực hiện câu lệnh insert into

```
//tạo một đối tượng SQLCommand
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn; //gán Connection cho SQLCommand
cmd.CommandText = "insert into LoaiSach values('" +
tbMaloai.Text + "',N'" + tbTenLoai.Text + "')";
cmd.ExecuteNonQuery();//thực hiện câu lệnh
```

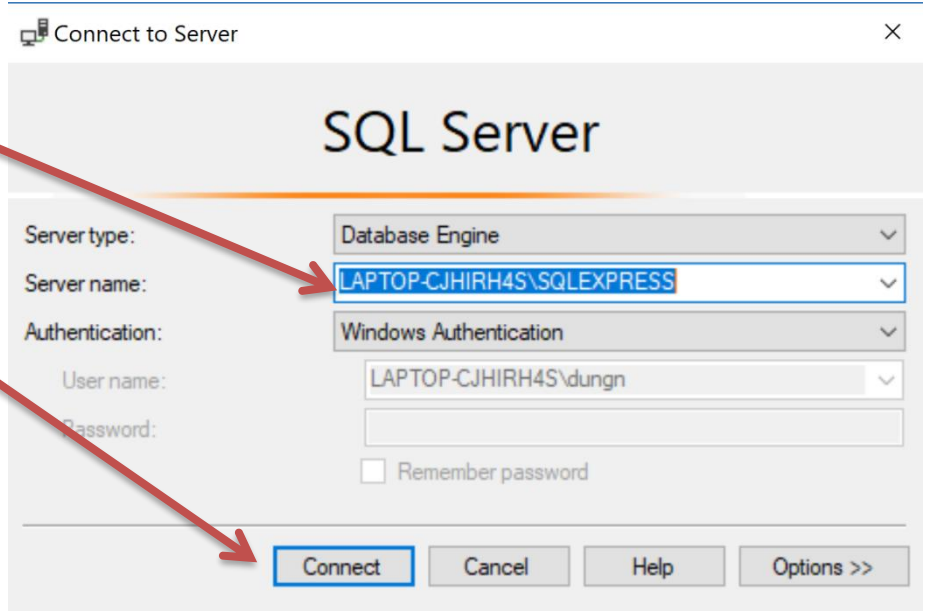
SqlDataReader

- SqlDataReader là đối tượng được thiết kế nhằm lấy dữ liệu từ CSDL một cách nhanh nhất.
- SqlDataReader được tạo ra bằng cách gọi phương thức ExecuteReader() của SqlCommand.
- Chỉ có thể đọc kết quả chứa trong đối tượng SqlDataReader từ đầu đến cuối và không thể thay đổi nó.
  - Khi gọi phương thức SqlDataReader.Read() thì nó sẽ load hàng dữ liệu kế tiếp. Nếu không có hàng nào được load thì phương thức Read() sẽ trả về false để báo hết dữ liệu.
  - SqlDataReader cung cấp các phương thức để lấy thứ tự của cột đó và trả về dữ liệu của cột; như GetString(), GetValue() …
- Không thể thực hiện SqlCommand có cùng đối tượng SqlConnection với SqlDataReader đang được mở.
- Nên sử dụng phương thức Close() của SqlDataReader sau khi đã lấy xong dữ liệu.

Tạo CSDL bằng công cụ Microsoft SQL Server Management Studio

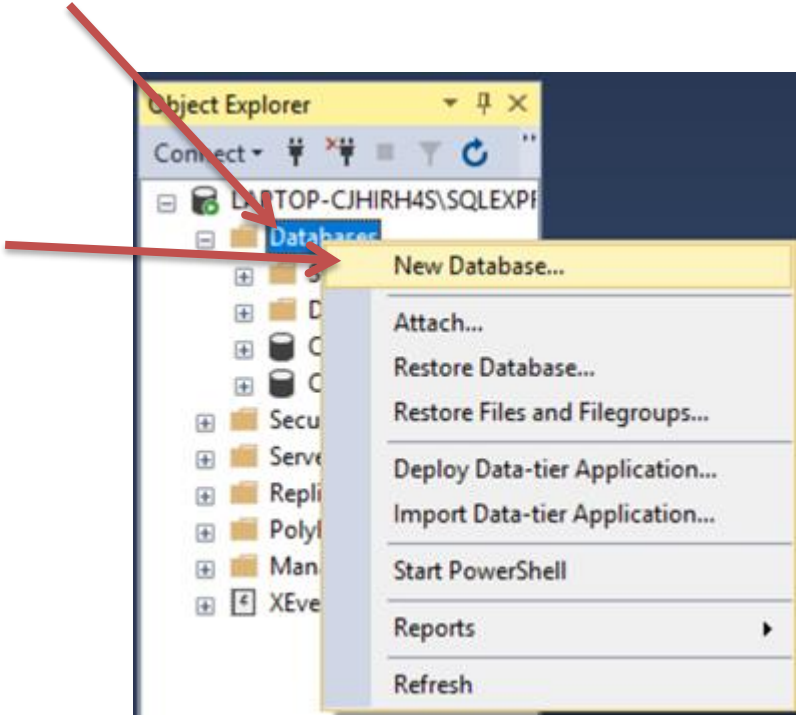
Khởi động SQL Server Management Studio

- B1: Chọn Server name rồi bấm vào nút Connect để kết nối đến SQL Server



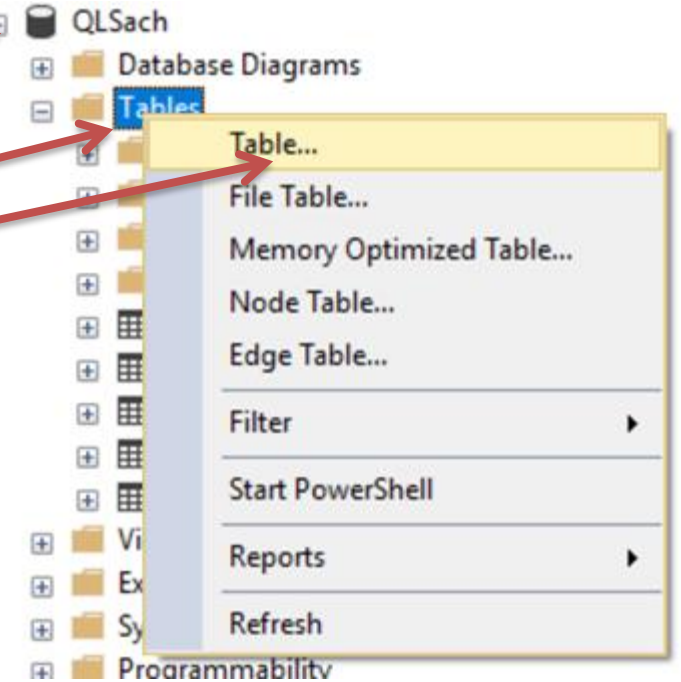
Tạo mới một cơ sở dữ liệu

- B2: Bấm chuột phải vào Database rồi chọn New Database để tạo mới một CSDL



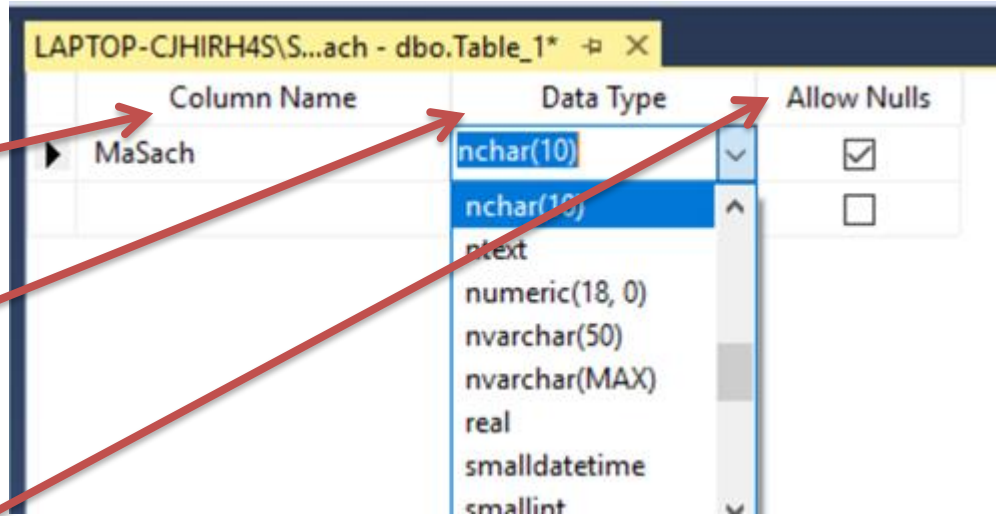
Tạo bảng

- Chọn cơ sở dữ liệu
- Chuột phải vào Table
- Chọn Table



Tạo bảng

- Khai báo các trường cho bảng
- Column Name: tên trường
- Data Type: kiểu dữ liệu
- Allow Nulls: cho phép rỗng hoặc không

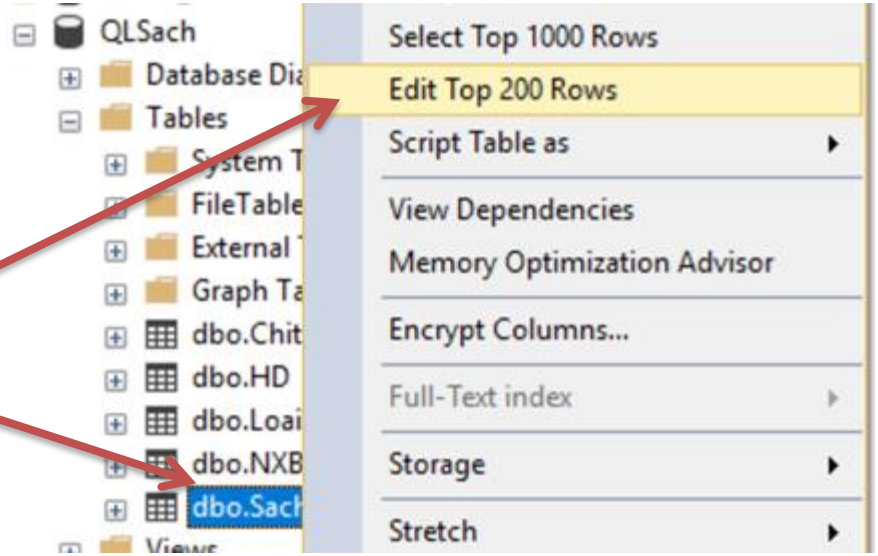


Tạo bảng

- Chú ý: Khi tạo bảng có quan hệ thì chỉ nên lưu trữ mã nhằm tránh dư thừa dữ liệu
- Ví dụ: trong phần mềm quản lý cửa hàng sách muốn quản lý sách theo thể loại, mỗi thể loại gồm rất nhiều đầu sách khác nhau => để tránh việc lặp đi lặp lại tên thể loại trong bảng sách thì sẽ tách ra làm 2 bảng Loại sách (mã loại, tên loại) và Sách(mã sách, mã loại, tên sách, ….) trong đó có trường mã loại được lưu trữ ở cả 2 bảng giúp kết nối tương ứng sách nào thuộc loại nào.

Chỉnh sửa dữ liệu trong bảng bằng cách trực tiếp

- Chuột phải vào tên bảng cần chỉnh sửa
- Chọn Edit …



Chỉnh sửa dữ liệu trong bảng bằng cách trực tiếp

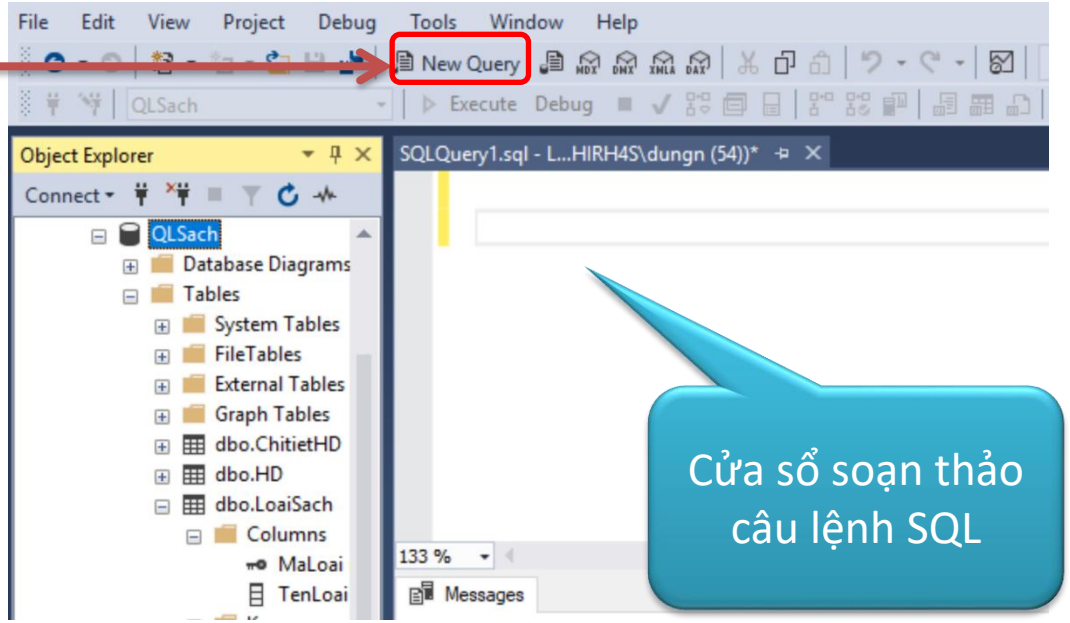
LAPTOP-CJHIRH4S\...QLSach - dbo.Sach | LAPTOP-CJHIRH4S\S...ach - dbo.Table_1*

	MaSach	MaLoai	TenSach	SoLuong	MaNXB	Dongia
	SV01	M02	Văn học lớp 8	5	NXBGD	15000
▶*	NULL	NULL	NULL	NULL	NULL	NULL

- Thực hiện thêm sửa xóa trực tiếp vào bảng

Sử dụng câu lệnh SQL để làm việc với cơ sở dữ liệu

- Chọn New Query để mở ra cửa sổ soạn thảo câu lệnh SQL



Một số câu lệnh SQL thường dùng

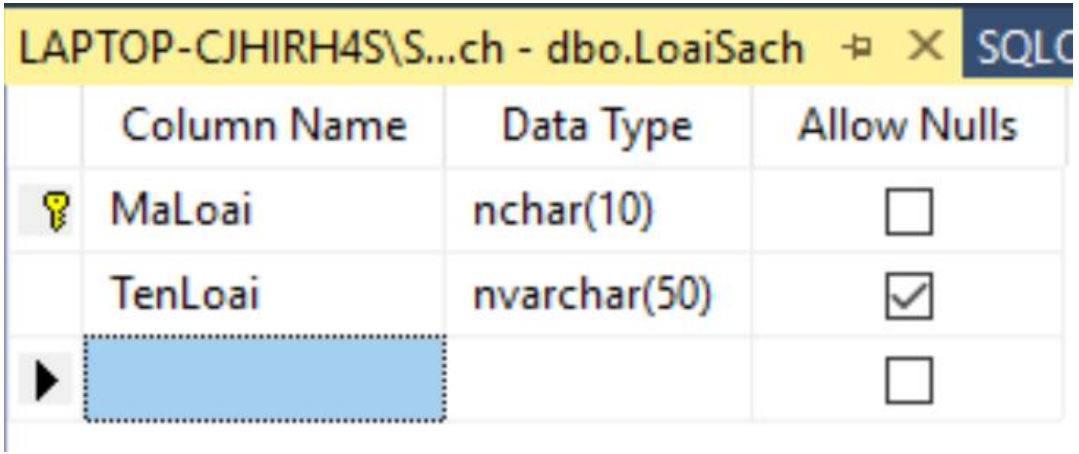
- INSERT INTO…
- UPDATE
- DELETE
- SELECT

INSERT INTO…

- Dùng để thêm dữ liệu vào bảng
- Cú pháp: Insert into <tên bảng> values(<các giá trị tương ứng với các trường>)

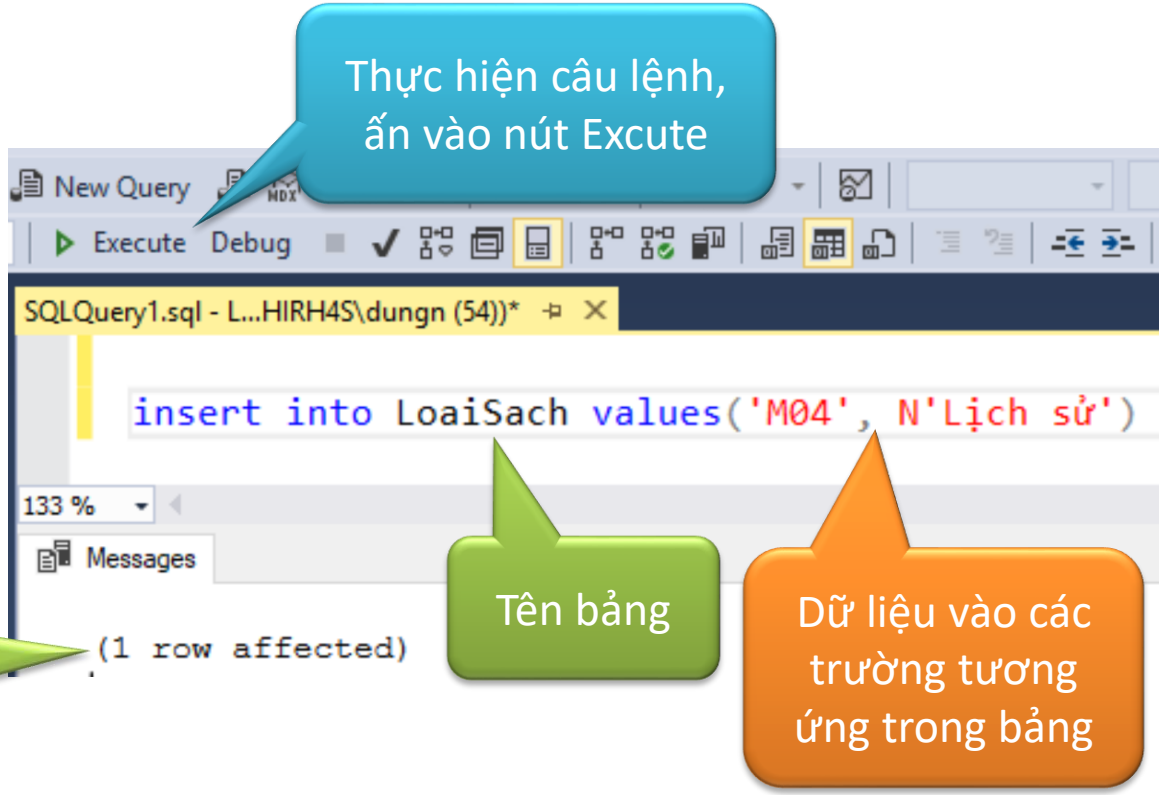
INSERT INTO…

- Ví dụ: Bảng Loại sách có trường mã loại sách và tên loại sách kiểu nchar và nvarchar



INSERT INTO…

=> Để thêm dữ liệu cho bảng Loại sách sử dụng câu lệnh như hình bên



UPDATE

- Dùng để sửa dữ liệu trong bảng
- Cú pháp: Update <tên bảng> set <tên trường> = <giá trị mới> where <điều kiện>
- Ví dụ: sửa tên loại trong bảng loại sách tại dòng có mã M04 thành Sinh học

```
Update LoaiSach set TenLoai = N'Sinh học' where MaLoai = 'M04'
```

DELETE

- Dùng để xóa dòng dữ liệu khỏi bảng
- Cú pháp: Delete <tên bảng> where <điều kiện>
- Ví dụ: xóa dòng có mã là M04 trong bảng Loại sách:

```
Delete LoaiSach where MaLoai = 'M04'
```

SELECT

- Dùng để truy vấn dữ liệu từ cơ sở dữ liệu
- Cú pháp:

  SELECT <danh sách các trường>
  FROM <danh sách các bảng>
  WHERE <điều kiện>
- Nếu lấy tất cả các trường trong bảng thì thay danh sách các trường bằng dấu *

SELECT

- Ví dụ:
- Lấy tất cả dữ liệu trong bảng Loại sách thì dùng câu lệnh:

```
select * from LoaiSach
```

- Kết quả:

MaLoai	TenLoai
M01	Toán học
M02	Văn học
M03	Anh Ngữ

SELECT

- Ví dụ:
- Muốn tìm các sách thuộc loại toán học thì dùng câu lệnh:

```
select * from Sach where MaLoai = 'M01'
```

SELECT

- Ví dụ truy vấn dữ liệu từ nhiều bảng:
- Muốn đưa ra chi tiết một đơn hàng (tên sách, số lượng, giá bán, thành tiền) trong khi bảng chi tiết đơn hàng chỉ chứa mã sách => dùng lệnh:

```
select S.TenSach, CT.SoLuong, CT.Giaban,
            CT.SoLuong*CT.Giaban as Thanhtien
from Sach as S, ChitietHD as CT
where S.MaSach = CT.MaSach
```

SELECT

- Trong đó:
  - Từ khóa as dùng để thay thế tên trường hoặc tên bảng bằng một tên mới
  - Có thể sử dụng các toán tử +, - , * , / trong câu lệnh SELECT

SELECT

- Các hàm có thể dùng trong câu lệnh SELECT:
- Sum: tính tổng giá trị các bản ghi
- Max: tìm giá trị lớn nhất trong các bản ghi
- Min: tìm giá trị nhỏ nhất trong các bản ghi
- Count: đếm số lượng các bản ghi
- Avg: tính trung bình cộng giá trị các bản ghi

SELECT

- Ví dụ:
- Tính tổng tiền phải trả của đơn hàng có mã HD01

```
select sum(soluong*giaban) as 'thành tiền'
from ChitietHD
where MaHD = 'HD01'
```