



Lập Trình WINDOWS

KẾT NỐI VÀ THAO TÁC VỚI CƠ SỞ DỮ LIỆU

ThS. Hồ Ngọc Trung Kiên



Nội Dung

❖ Mô hình kết nối

- Đối tượng Connection
- Đối tượng Command
- Đối tượng DataReader

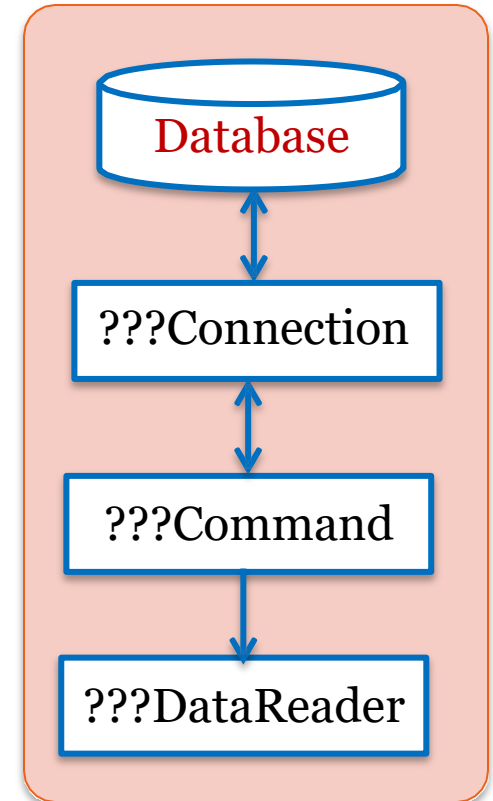
❖ Mô hình phi kết nối

- Đối tượng DataAdapter
- Đối tượng DataSet
- Đối tượng DataTable, DataRow, DataColumn



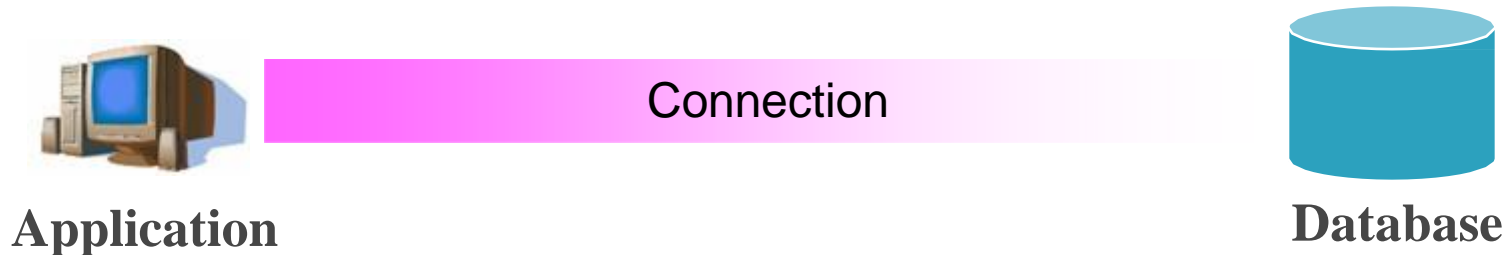
1. Mô hình kết nối

- ❖ Các đối tượng để thao tác với cơ sở dữ liệu trong mô hình kết nối:
 - Connection: Dùng để thiết lập kết nối đến cơ sở dữ liệu.
 - Command: Dùng để thực hiện các câu lệnh truy vấn SQL để lấy về dữ liệu (Select) hoặc cập nhật dữ liệu (Insert, Update, Delete).
 - DataReader: Dùng để duyệt các bản ghi theo chiều tiến và chỉ đọc.



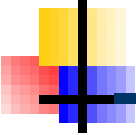
1.1. Đối tượng Connection

❖ **Chức năng:** Thiết lập kết nối đến cơ sở dữ liệu



Thuộc tính & Phương thức

- **ConnectionString:** chuỗi kết nối DataSource
- **State:** trạng thái kết nối
- **Open():** thiết lập kết nối đến DS
- **Close():** đóng kết nối với DS
- **Dispose():** hủy kết nối



1.1. Đối tượng Connection

```
???Connection con;  
string strcon= “Chuỗi kết nối”;  
con = new ???Connection()  
con.ConnectionString = strcon;  
con.Open();  
  
//xử lý trong quá trình kết nối  
...  
con.Close();  
con.Dispose();
```



1.1. Đối tượng Connection

❖ Kết nối với cơ sở dữ liệu Ms Access:

```
OleDbConnection con;// Khai báo
string strcon;
strCon = @"Provider=Microsoft.Jet.OleDb.4.0;Data Source =" +
Application.StartupPath + "\\QLSV.mdb"; //Access 2003
Hoặc strCon = @"Provider=Microsoft.Ace.OleDb.12.0;
Data Source =" + Application.StartupPath + "\\QLSV.accdb" ; //2007
con = new OleDbConnection(strCon) ;
con.Open() ;
```

```
string constr = @"Provider=Microsoft.Jet.OleDb.4.0;Data Source = D:\QLSV.mdb";
string constr = @"Provider=Microsoft.Ace.OleDb.12.0;Data Source = QLSV.accdb";
string constr = @"Provider=Microsoft.Ace.OleDb.12.0;Data Source =" +
Application.StartupPath + "\\QLSV.accdb";
```



1.1. Đối tượng Connection

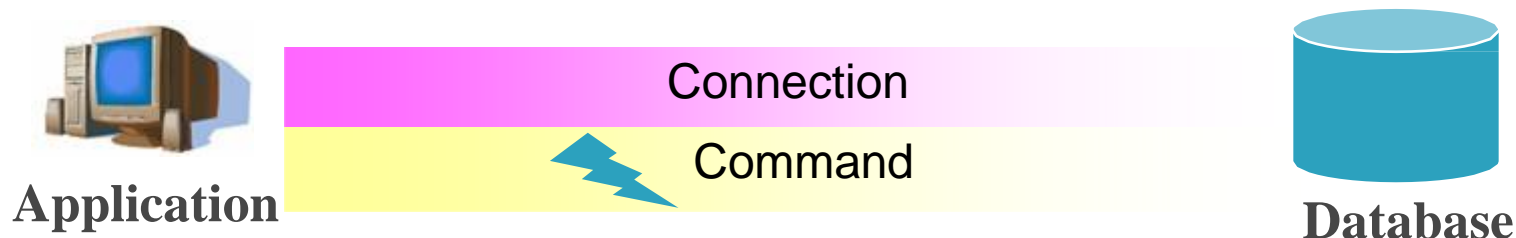
❖ *Kết nối với cơ sở dữ liệu Sql Server:*

```
SqlConnection con;// Khai báo
String strcon;
strCon=
"Server=ServerName;Database=DBName;uid=userName;pwd=password;";
con = new SqlConnection(strCon);
con.Open();
```

```
strCon=@"Data Source=minh;Database=QLSV;User ID=sa;Password=sa";
strCon=@"Server=minh;Initial Catalog=QLSV;uid=sa;pwd=sa";
strCon=@"Data Source=minh;Database=QLSV;Integrated
Security=true";           //Integrated Security=SSPI
```

1.2. Đối tượng Command

- ❖ **Chức năng:** Dùng để thực hiện các câu lệnh SQL thao tác với CSDL như: **Insert, Update, Select, Delete...**



Thuộc tính & Phương thức

- **Connection:** kết nối để thực hiện lệnh
- **CommandType:** loại câu lệnh (Text, TableDirect, StoredProc)
- **CommandText:** câu lệnh cần thực hiện
- **Parameters:** xác định các tham số lệnh
- **ExecuteScalar():** thực hiện câu lệnh và trả về giá trị đơn
- **ExecuteNonQuery():** gọi các lệnh SQL, store, trả về số row bị tác động (**Insert, Update, Delete...**)
- **ExecuteReader():** thực hiện lệnh và trả về DataReader



Tạo đối tượng Command

- ❖ Tạo Command từ phương thức tạo dựng

```
???Command cmd = new ???Command("SELECT * FROM  
sinhvien", con);
```

- ❖ Tạo Command từ phương thức CreateCommand của đối tượng Connection

```
???Command cmd = con.CreateCommand();  
cmd.CommandText = "SELECT * FROM sinhvien";
```

Tạo đối tượng Command ...

- ❖ Tạo Command bằng cách đặt các thuộc tính sau khi khai báo

```
???Command cmd = new ???Command();  
cmd.CommandType = CommandType.Text;  
cmd.CommandText = "SELECT * FROM sinhvien";  
cmd.Connection = con;
```

```
SqlCommand cmd = new SqlCommand();  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = "Get_all_SV";  
cmd.Connection = con;
```

Tạo đối tượng Command ...

- ❖ Tạo Command bằng cách đặt các thuộc tính sau khi khai báo (tt)

```
???Command cmd = new ???Command();  
cmd.CommandText = "SELECT * FROM sinhvien  
WHERE MaLOP=' '+txtMaLOP.text+' '";  
cmd.Connection = con;
```

```
SqlCommand cmd = new SqlCommand();  
cmd.CommandType = CommandType.TableDirect;  
cmd.CommandText = "SinhVien"; //Bảng sinh viên  
cmd.Connection = con;
```

Thực thi câu lệnh Command

❖ **ExcuteReader()**: Trả về một tập các bản ghi.

- Nó tương đương với một Recordset
- Thường được sử dụng để thực thi các câu lệnh truy vấn như **SELECT**. (kết quả có thể lưu trữ trong đối tượng DataReader để thao tác)

```
...  
???Command cmd = new ???Command("SELECT * FROM Sinhvien",  
con);  
con.Open();  
???DataReader dr = cmd.ExecuteReader();  
//dr là biến DataReader, sẽ tìm hiểu sau  
...  
con.Close();
```

Thực thi câu lệnh Command

❖ **ExcuteScalar():** Trả về một giá trị đơn.

- Có thể là phần tử cột đầu tiên hàng đầu tiên trong bảng kết quả.
- Phương thức này thường được sử dụng thực hiện câu lệnh truy vấn **SELECT** mà kết quả trả về là một giá trị (**SELECT Count(*) FROM sinhvien**).

```
...  
???Command cmd = con.CreateCommand();  
cmd.CommandText = "SELECT count(*) FROM Sinhvien";  
con.Open();  
int count = (int) cmd.ExecuteScalar();  
//count = số bản ghi trong table Sinh viên  
...  
con.Close();
```

Thực thi câu lệnh Command

- ❖ **ExcuteNonQuery()**: Được sử dụng để thực thi các câu lệnh truy vấn hành động: **INSERT, UPDATE, DELETE...**

```
...  
???Command cmd = new ???Command();  
cmd.CommandType = CommandType.Text;  
cmd.CommandText = "INSERT Into Sinhvien Values(007,,Nguyễn Văn  
A','10/02/1982',1,null)";  
cmd.Connection = con;  
con.Open();  
cmd.ExecuteNonQuery();  
...  
con.Close();
```

Truyền tham số cho câu lệnh

...

```
SqlCommand cmd = new SqlCommand();  
cmd.CommandType = CommandType.Text;  
cmd.CommandText = "INSERT Into Sinhvien  
    Values(@MS,@HT,@NS,null,null)";  
cmd.Parameters.Add("@MS", SqlDbType.Int);  
cmd.Parameters.Add("@HT", SqlDbType.NVarChar,30);  
cmd.Parameters.Add("@NS", SqlDbType.DateTime);  
cmd.Parameters["@MS"].Value = 12;  
cmd.Parameters["@HT"].Value = "Nguyễn Quốc Cường";  
cmd.Parameters["@NS"].Value = new DateTime(1982,02,10);  
cmd.Connection = con;  
con.Open();  
cmd.ExecuteNonQuery();  
con.Close();
```

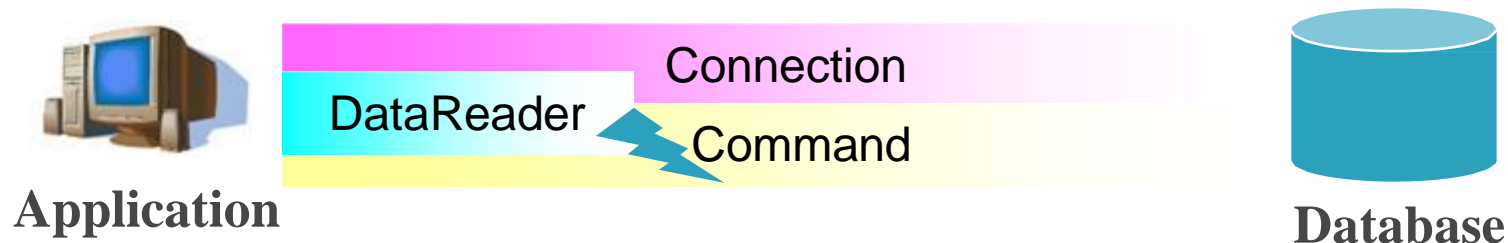
Câu lệnh được
truyền tham số

Khai báo
tham số

Truyền giá trị
cho tham số

1.3. Đối tượng DataReader

- ❖ **Chức năng:** Dùng để đón nhận kết quả trả về từ đối tượng Command thông qua phương thức **ExecuteReader()**.



Thuộc tính & Phương thức

- **Read():** đọc một mẫu tin
- **NextResult():** Chuyển tới record kế tiếp
- **dr[i]** or **dr["FieldName"]:** truy xuất đến cột **i** của mẫu tin được đọc.
- **Close():** đóng

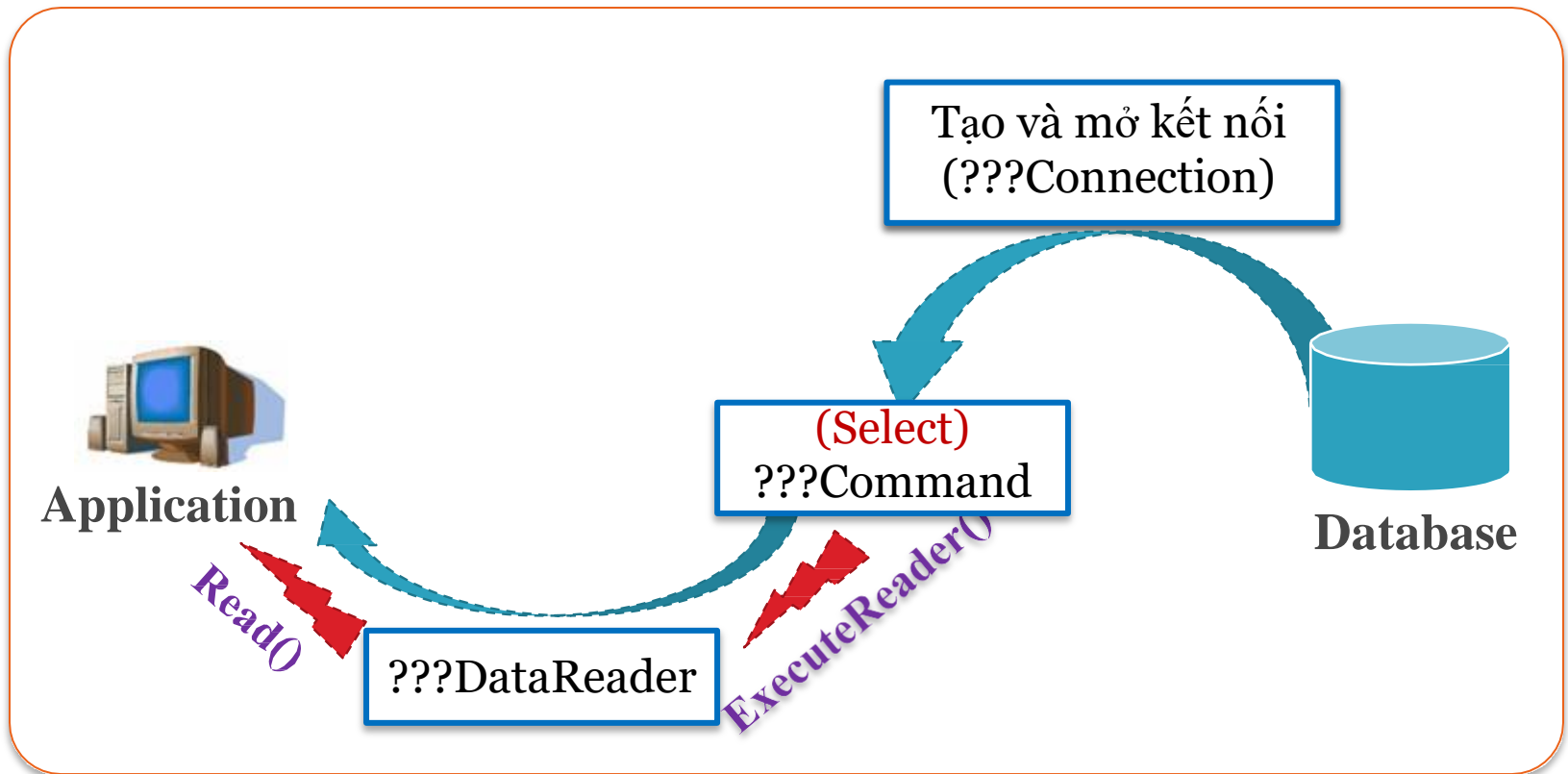
DataReader hoạt động theo cơ chế truy xuất tuần tự và chỉ đọc dữ liệu

1.3. Đối tượng DataReader...

❖ Ví dụ

```
...  
???Command cmd = new ???Command("SELECT * FROM Sinhvien",  
con);  
con.Open();  
???DataReader dr = cmd.ExecuteReader();  
while (dr.Read())  
    listBox1.Items.Add(dr["Hoten"]);  
dr.Close();  
con.Close();
```

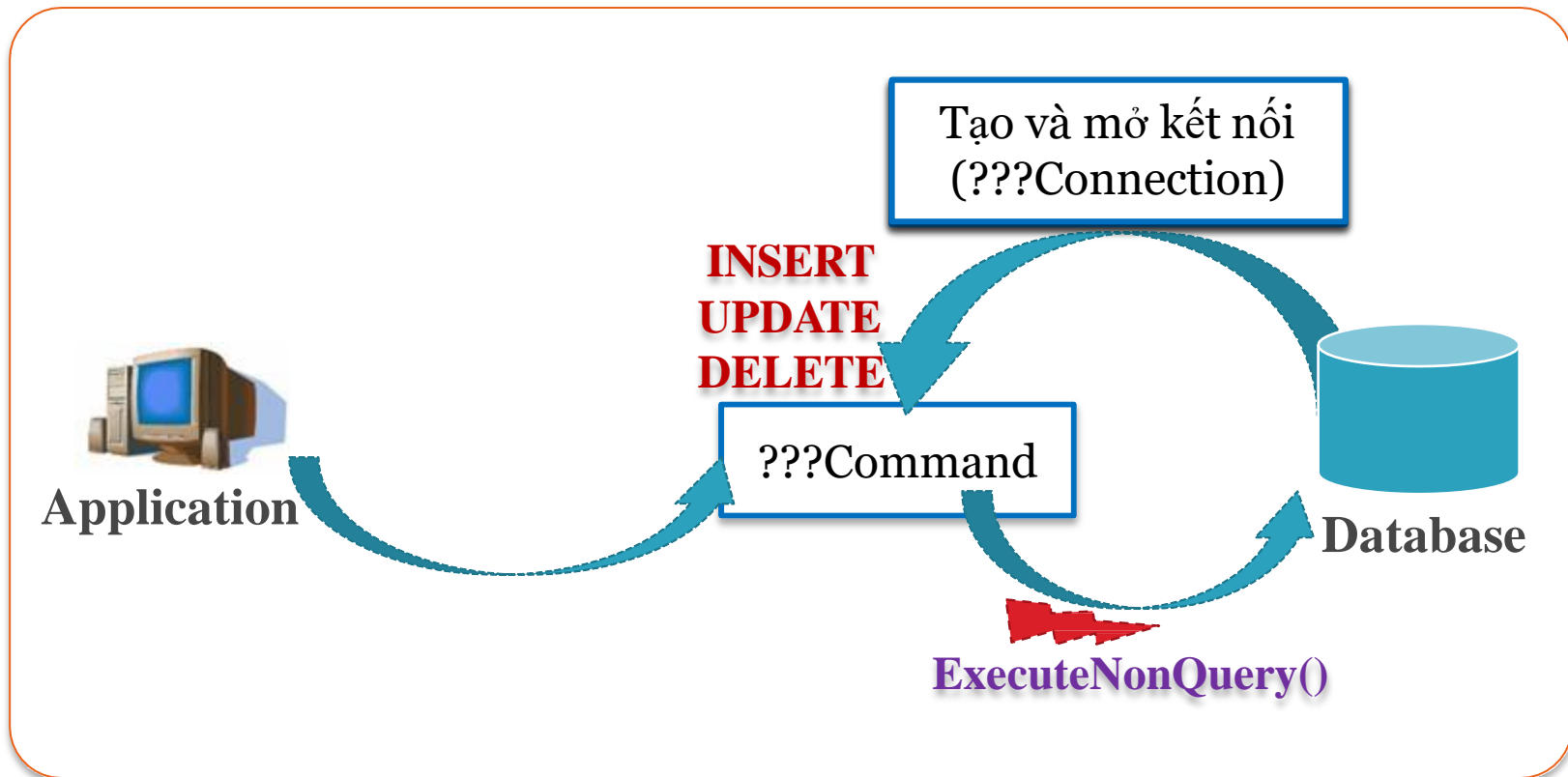
Lấy dữ liệu trong mô hình kết nối



Ví dụ

```
string constr = @"Server=Admin;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText="SELECT * FROM Sinhvien";  
cmd.Connection=con;  
SqlDataReader dr = cmd.ExecuteReader();  
while (dr.Read())  
    listBox1.Items.Add(dr["Hoten"]);  
dr.Close();  
con.Close();
```

Thao tác dữ liệu trong mô hình kết nối



Ví dụ: INSERT

```
string constr = @"Server=Admin;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText="INSERT Into sinhvien Values('' + this.Txtmalop.Text  
+ ''','' + this.TxtmaSV.Text + ''','' + this.txtTen.Text + '')";  
cmd.Connection=con;  
cmd.ExecuteNonQuery();  
cmd.Dispose();  
con.Close();  
con.Dispose();
```



Ví dụ: UPDATE

```
string constr = @"Server=minh;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText= "UPDATE sinhvien Set malop='" +  
this.Txtmalop.Text + "',masvien='" + this.TxtmaSV.Text + "', tensvien='" +  
this.txtTen.Text + "' WHERE masvien='" + this.TxtmaSV.Text + '"";  
cmd.Connection=con;  
cmd.ExecuteNonQuery();  
con.Close();
```

Ví dụ: DELETE

```
string constr = @"Server=Admin;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText= "DELETE FROM sinhvien WHERE masvien='" +  
this.TxtmaSV.Text + "'";  
cmd.Connection=con;  
cmd.ExecuteNonQuery();  
con.Close();
```



Ví dụ: Gọi thủ tục Stored Procedure

❖ Nội dung thủ tục nội tại (insert_student)

insert_student

```
CREATE PROCEDURE insert_student
```

```
(
```

```
    @tensv nvarchar(30),
```

```
    @ngaysinh datetime,
```

```
    @diachi nvarchar(50),
```

```
)
```

```
AS
```

```
    INSERT Into hocsinh VALUES(@tensv, @ngaysinh, @diachi)
```

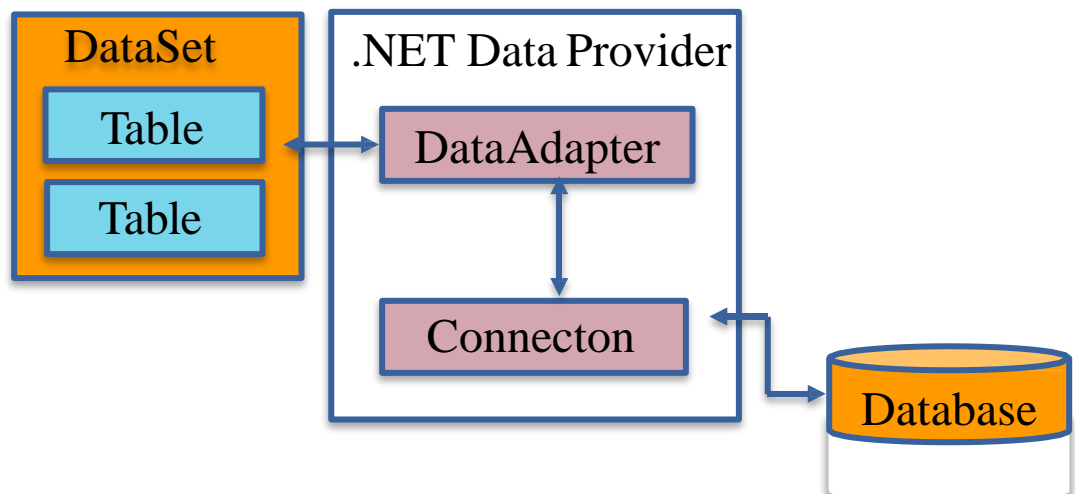

Ví dụ: Gọi thủ tục Stored Procedure ...

```
string constr =  
    @"Server=Admin;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = "insert_student";  
cmd.Parameters.Add("@tensv", SqlDbType.NVarChar, 50);  
cmd.Parameters["@tensv"].Value = "Nguyễn Quốc Cường";  
cmd.Parameters.Add("@ngaysinh", SqlDbType.DateTime);  
cmd.Parameters["@ngaysinh"].Value = new DateTime(1982, 02, 10);  
cmd.Parameters.Add("@diachi", SqlDbType.NVarChar, 50);  
cmd.Parameters["@diachi"].Value = "Bình Dương";  
cmd.Connection=con;  
cmd.ExecuteNonQuery();  
con.Close();
```



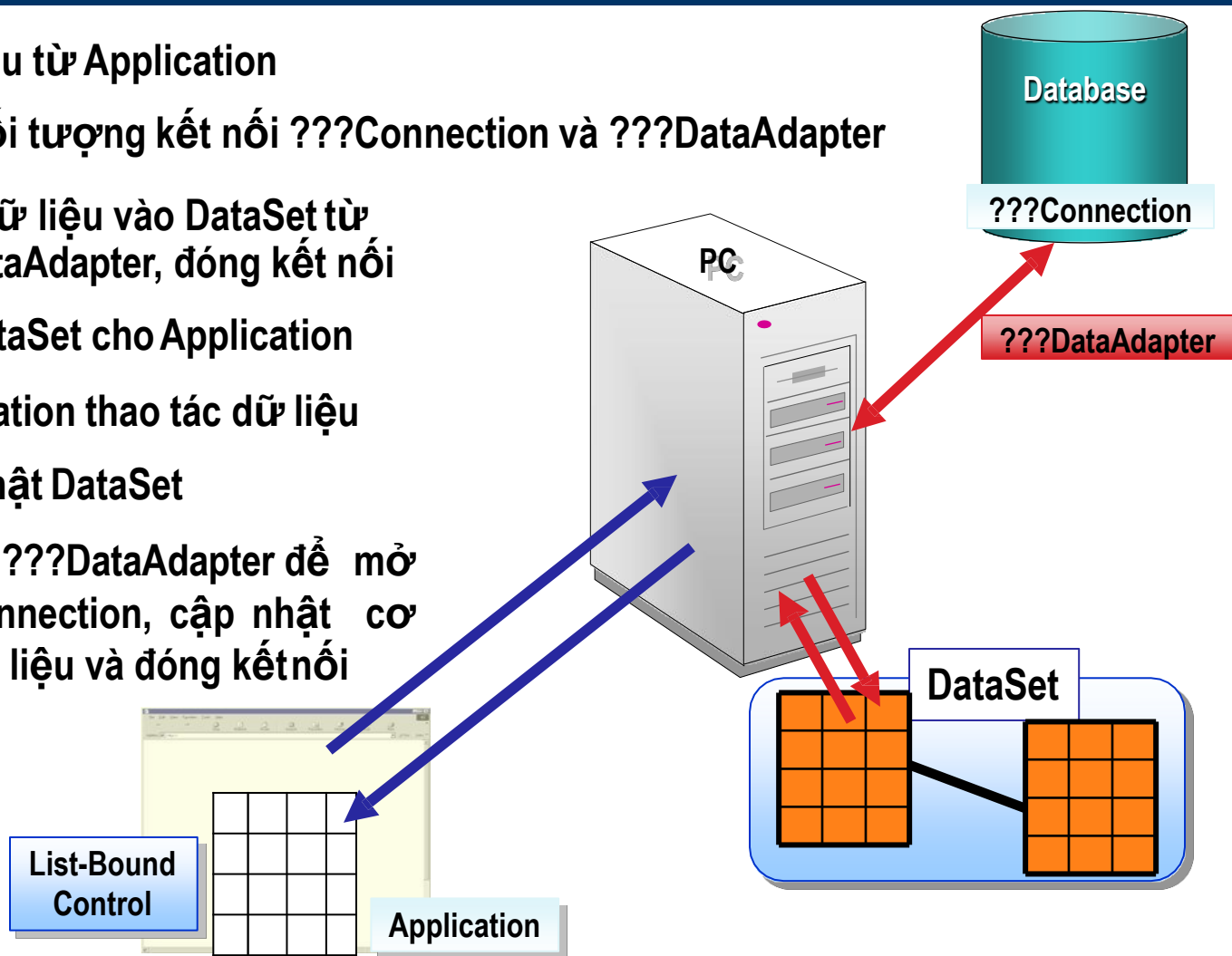
2. Mô hình phi kết nối

- ❖ Các đối tượng để thao tác với cơ sở dữ liệu trong mô hình phi kết nối:
 - Connection: Dùng để thiết lập kết nối đến cơ sở dữ liệu (có thể có hoặc không).
 - DataAdapter: Là cầu nối giữa DataSet và Database .
 - DataSet: Giống như một kho chứa các bảng (Table) hay là một Database thu nhỏ.
 - DataTable,...



2. Mô hình phi kết nối ...

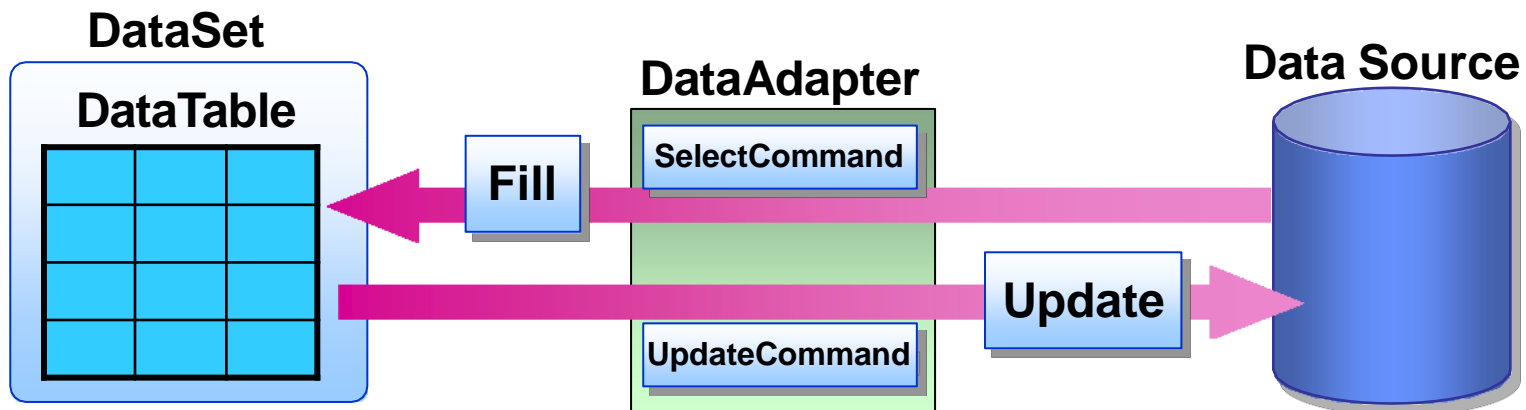
1. Yêu cầu từ Application
2. Tạo đối tượng kết nối ???Connection và ???DataAdapter
3. Điền dữ liệu vào DataSet từ ???DataAdapter, đóng kết nối
4. Trả DataSet cho Application
5. Application thao tác dữ liệu
6. Cập nhật DataSet
7. Sử dụng ???DataAdapter để mở ???Connection, cập nhật cơ sở dữ liệu và đóng kết nối



2.1. Đối tượng DataAdapter

Thuộc tính & Phương thức

- **SelectCommand, InsertCommand, UpdateCommand, DeleteCommand**: thực hiện các thao tác select, insert, update, delete dữ liệu.
- **Fill(DataSet)**: lấy dữ liệu từ Data Source thông qua lệnh SelectCommand đổ vào Data Set
- **Update(DataSet)**: cập nhật dữ liệu trong DataSet vào DataSource





2.1. Đối tượng DataAdapter ...

- ❖ **Ví dụ 1:** Tạo DataAdapter thông qua chuỗi truy vấn và chuỗi kết nối.

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
DataSet ds = new DataSet();  
  
da.Fill(ds);  
// thao tác trên dataset  
...  
da.Update(ds);
```

2.1. Đối tượng DataAdapter ...

❖ **Ví dụ 2:** Tạo DataAdapter thông qua chuỗi truy vấn và đối tượng Connection.

```
string constr = @"Server=minh;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();  
  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", con);  
DataSet ds = new DataSet();  
  
da.Fill(ds);  
// thao tác trên dataset  
...  
da.Update(ds);  
con.Close();
```

2.1. Đối tượng DataAdapter ...

❖ Ví dụ 3: Tạo DataAdapter thông qua đối tượng Command

```
string constr = @"Server=minh;Database=QLSV;UID=sa;Password=sa";  
SqlConnection con = new SqlConnection(constr);  
con.Open();
```

```
SqlCommand cmd = new SqlCommand();  
cmd.CommandText="SELECT * FROM Sinhvien";  
cmd.Connection=con;
```

```
SqlDataAdapter da = new SqlDataAdapter();  
da.SelectCommand = cmd;  
//SqlDataAdapter da = new SqlDataAdapter(cmd);
```

```
DataSet ds = new DataSet();  
da.Fill(ds, 'tblsinhvien');
```

...

```
con.Close();
```



2.2. DataSet

32

- ❖ Là phần cơ sở dữ liệu được lưu trữ trong bộ nhớ (in-memory database)
- ❖ Gồm tập các đối tượng DataTable mỗi DataTable diễn hình tương đương với một table hoặc là một view, ta có thể thiết lập mối quan hệ giữa chúng thông qua đối tượng DataRelation.
- ❖ Dữ liệu trong DataSet có thể nhất quán bằng cách sử dụng đối tượng UniqueConstraint và ForeignKeyConstraint

2.2. DataSet

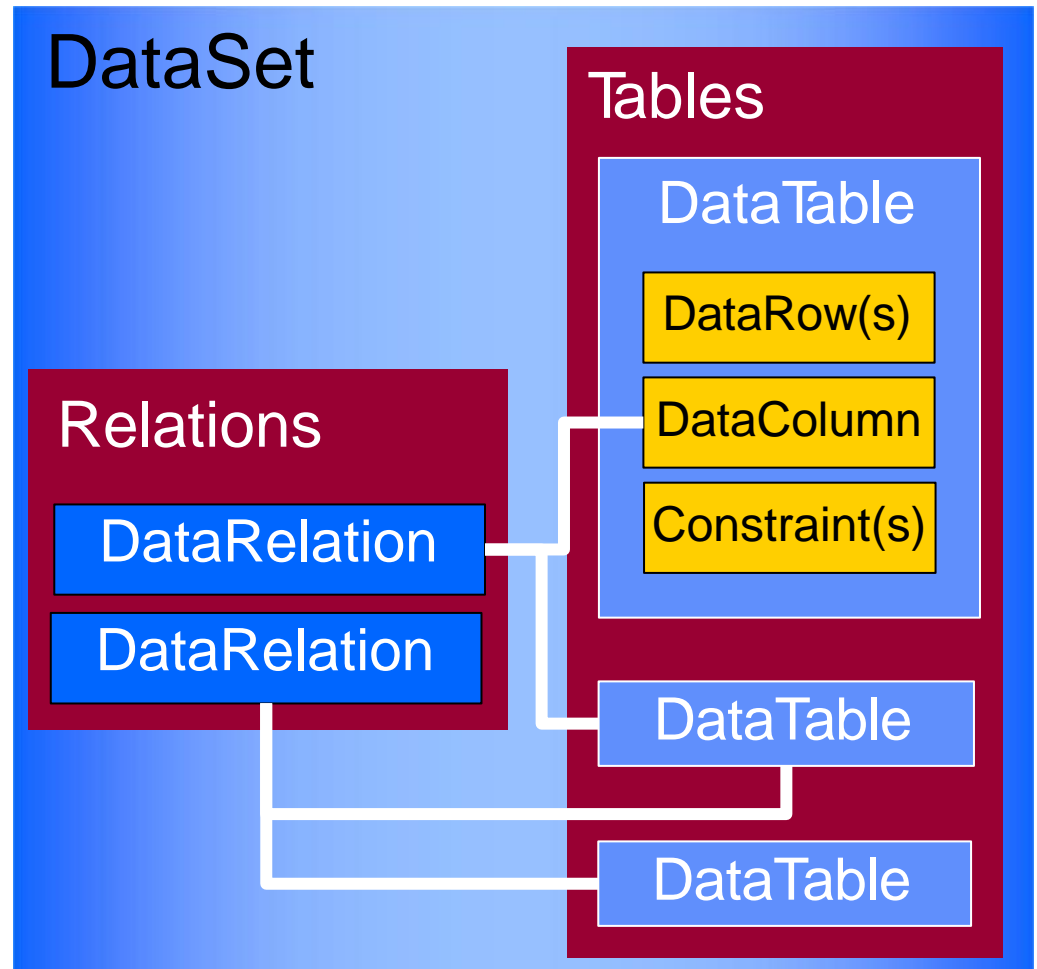
DataSet vs. DataReader

DataSet	DataReader
Read/write access to data	Read-only
Includes multiple tables from different databases	Based on one SQL statement from one database
Disconnected	Connected
Bind to multiple controls	Bind to one control only
Forward and backward scanning of data	Forward-only
Slower access	Faster access
Supported by Visual Studio .NET tools	Manually coded

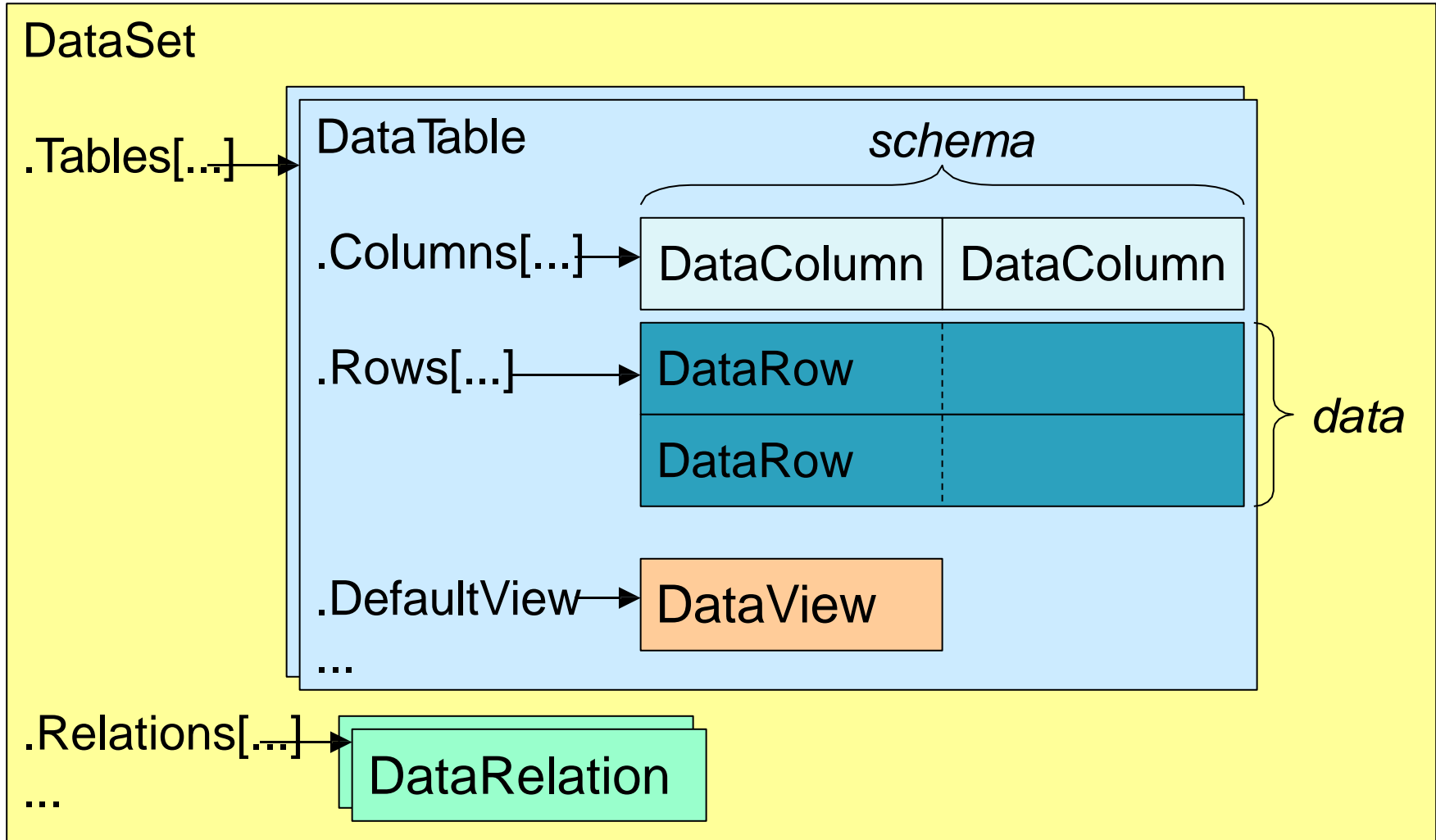
2.2. DataSet

❖ Các thành phần

- Tables
- Relations



2.2. DataSet





2.2. DataSet

Thuộc tính & Phương thức

- **Tables**: trả về danh sách các Table chứa trong DataSet.
- **Tables[i]** or **Tables["tblName"]** : Trả về đối tượng Table thứ i hoặc Table có tên tblName.
- **Relations**: trả về danh sách các quan hệ giữa các Table trong DataSet
- **Tables.Count**: trả về số lượng các Table chứa trong DataSet
- **Tables.Add("tblName")**: thêm Table tblName vào DataSet



Ví dụ

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
DataSet ds = new DataSet();  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
da.Fill(ds, "sinhvien");  
SqlDataAdapter da1 = new SqlDataAdapter("Select * From lop", constr);  
da1.Fill(ds);  
ds.Tables[0]; //ds.Tables["sinhvien"];  
ds.Tables[1];
```

2.3. DataTable

- ❖ DataTable: Là đối tượng để biểu diễn thông tin trong DataSet.

DataRow			
DataColumn			
ColName1	ColName2	ColNameN
Data11	Data12	...	Data1N
...
DataM1	DataM2	DataMN



2.3. DataTable...

Thuộc tính & Phương thức

- **TableName**: trả về tên Table
- **Columns**: trả về danh sách các cột
- **Columns.Count**: trả về số cột trong bảng
- **Rows**: danh sách các mẫu tin
- **Rows.Count**: trả về số dòng trong bảng
- **PrimaryKey**: danh sách các cột là khóa chính
- **NewRow()**: tạo một mẫu tin mới
- **Columns.Add(DataColumn)**: Thêm cột vào bảng
- **Rows.Add(DataRow)**: thêm một dòng vào bảng


```
DataTable dt = new DataTable();
```

2.3. DataColumn

❖ DataColumn: đại diện cho một cột trong bảng

- ColumnName: tên cột
- DataType: kiểu dữ liệu

DataTable



Column1	Column2	...	Column n

```
DataColumn dc = new DataColumn("ColumnName");  
DataTable dt=new DataTable();  
dt.Columns.Add(dc);
```


2.3. DataRow

- ❖ DataRow: đại diện cho mẫu tin trong bảng
 - RowState: trạng thái Added, Modified, Deleted,...
 - [i]: truy xuất đến cột i
 - Delete(): đánh dấu xóa mẫu tin

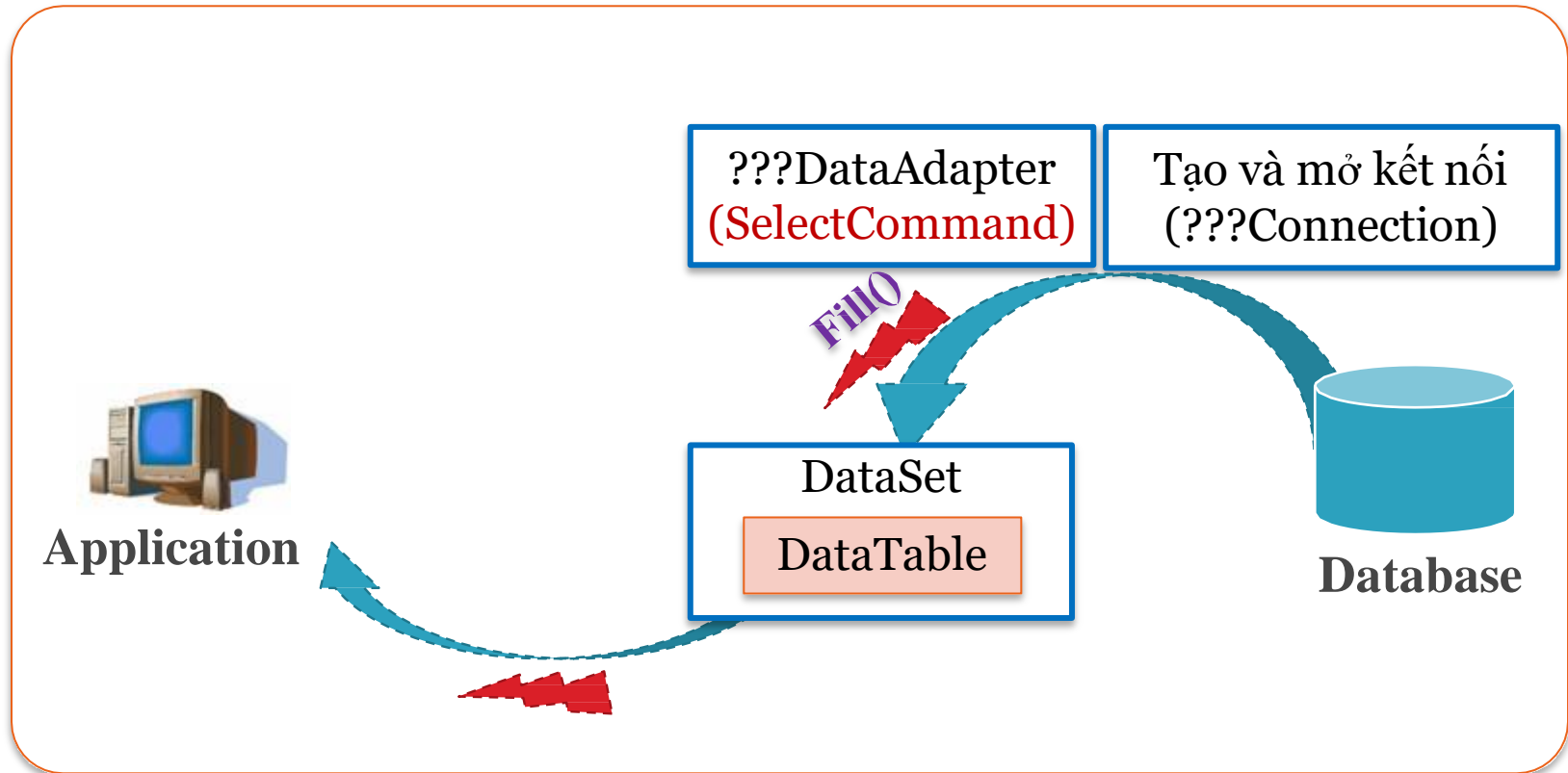
DataTable

Column1	Column2	...	Column n
abc	xyz		omt
...
...

Row

```
DataRow drw = new DataRow();
```


Lấy dữ liệu trong mô hình phi kết nối





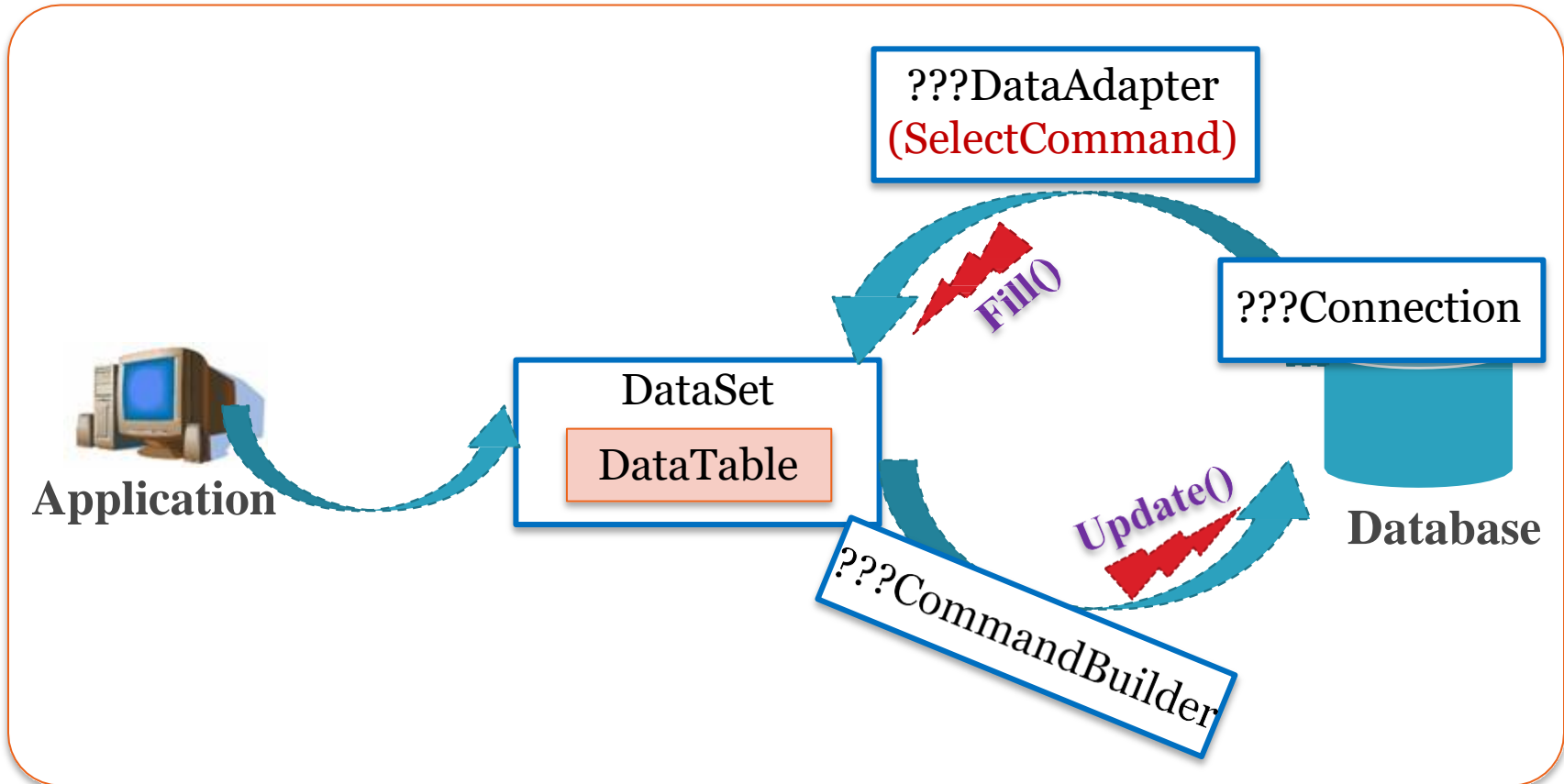
Ví dụ

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
DataSet ds = new DataSet();  
  
da.Fill(ds, "sinhvien");  
  
this.dataGridView1.DataSource = ds;  
this.dataGridView1.DataMember = "Sinhvien";  
  
this.dataGridView1.DataSource = ds.Tables["sinhvien"];
```



Thao tác dữ liệu trong mô hình phi kết nối

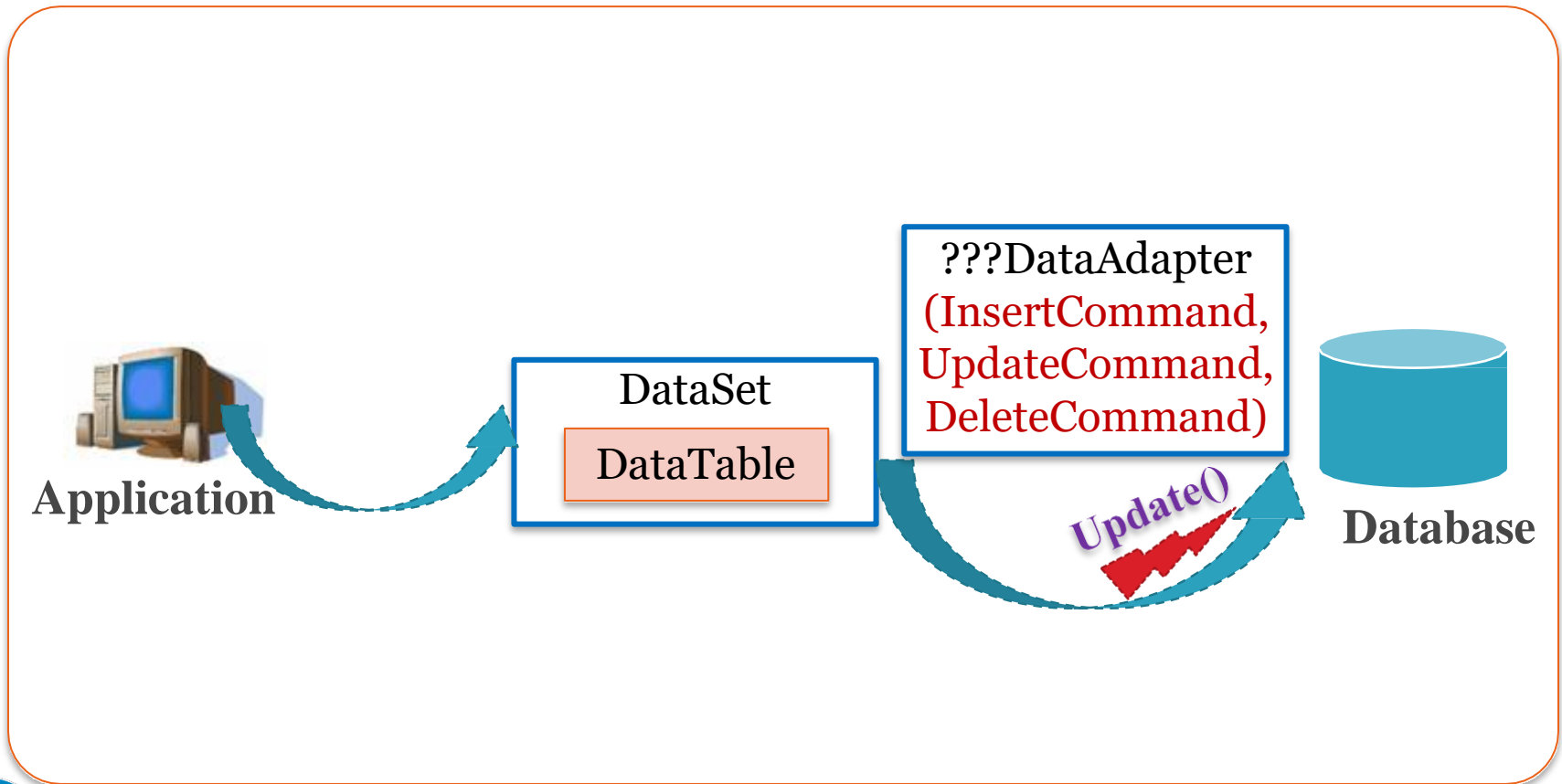
❖ Cách 1:



Thao tác dữ liệu trong mô hình phi kết nối...

45

❖ Cách 2:



Ví dụ: INSERT (c1)

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
DataSet ds = new DataSet();  
da.Fill(ds, "sinhvien");  
this.dataGridView1.DataSource = ds.Tables["sinhvien"];  
DataRow dr;  
DataTable dt;  
dt = ds.Tables[0];  
dr = dt.NewRow();  
dr[0] = this.txtMasv.Text;  
dr[1] = this.txtTensv.Text;  
...  
ds.Tables["sinhvien"].Rows.Add(dr); // ds.Tables[0].Rows.Add(dr);  
SqlCommandBuilder cb = new SqlCommandBuilder(da);  
da.Update(ds.Tables["sinhvien"]);
```

Ví dụ: INSERT (c2)

```
...
grvSinhvien.DataSource = ds.Tables["sinhvien"];
DataRow dr = ds.Tables[0].NewRow();
dr[0] = this.txtMasv.Text;    dr["Tensv"] = this.txtTensv.Text; ...
ds.Table[0].Rows.Add(dr);
SqlCommand cmd = new SqlCommand();
cmd.CommandText = @"Insert into sinhvien
                    values(@masv,@tensv,@ngaysinh,@gioitinh,@quequan,@malop)";
cmd.Parameters.Add("@masv", SqlDbType.NVarChar, 10, "masv");
...
cmd.Parameters.Add("@gioitinh", SqlDbType.Bit);
if (cbgioitinh.Text == "Nam")    cmd.Parameters["@gioitinh"].Value = 0;
else    cmd.Parameters["@gioitinh"].Value = 1;
...
cmd.Connection = con.connect();
SqlDataAdapter da = new SqlDataAdapter();
da.InsertCommand = cmd;
da.Update(ds.Tables["sinhvien"]);
```

Ví dụ: UPDATE

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
DataSet ds = new DataSet();  
da.Fill(ds);  
  
foreach (DataRow drw in ds.Tables[0].Rows)  
    drw["Ngaysinh"] = DateTime.Now;  
  
SqlCommandBuilder cb = new SqlCommandBuilder(da);  
da.Update(ds);
```


Ví dụ: DELETE

```
string constr = @"Server=minh;Database=QLSV;Integrated Security=true";  
  
SqlDataAdapter da = new SqlDataAdapter("Select * From Sinhvien", constr);  
DataSet ds = new DataSet();  
da.Fill(ds);  
  
DataTable dt = ds.Tables[0];  
DataRow[] rows = dt.Select("Ngaysinh < '1/1/1980'");  
foreach (DataRow drw in rows)  
    drw.Delete();  
SqlCommandBuilder cb = new SqlCommandBuilder(da);  
da.Update(ds);
```

Thảo luận

