



Khoa Công nghệ Thông tin – Đại học Đà Lạt

## Chương 4

### THỰC THI LỆNH THAO TÁC TRÊN CSDL (2)

## Mục tiêu

- Giúp sinh viên nắm rõ các vấn đề sau
  - Thực thi thao tác truy vấn dữ liệu và đọc kết quả trả về
  - Lưu trữ dữ liệu dùng Dataset
  - Trích lọc, sắp xếp dữ liệu với Dataview

## Nội dung

- Lớp SqlDataReader
- Tạo và sử dụng đối tượng DataReader
- Đọc dữ liệu từ DataReader
  - Lấy dữ liệu theo kiểu dữ liệu cụ thể
  - Đọc các giá trị **NULL**
- Thực thi nhiều lệnh truy vấn SQL

## Lớp SqlDataReader

- Dùng để **đọc tuần tự** các dòng lấy được từ cơ sở dữ liệu SQL Server
- Một số thuộc tính
  - Int **Depth** { get; }
  - Int **FieldCount** { get; }
  - Int **RecordsAffected** { get; }
  - Bool **IsClosed** { get; }
- Hầu hết các thuộc tính & phương thức này cũng được thiết kế cho lớp OleDbDataReader và OdbcDataReader.

## Lớp SqlDataReader

- Một số phương thức

RETURN TYPE	METHOD NAME	RETURN TYPE	METHOD NAME
bool	GetBoolean()	SqlBinary	GetSqlBinary()
byte	GetByte()	SqlBoolean	GetSqlBoolean()
long	GetBytes()	SqlByte	GetSqlByte()
char	GetChar()	SqlDateTime	GetSqlDateTime()
long	GetChars()	SqlDecimal	GetSqlDecimal()
string	GetDataTypeName()	SqlDouble	GetSqlDouble()
DateTime	GetDateTime()	SqlGuid	GetSqlGuid()
decimal	GetDecimal()	SqlInt16	GetSqlInt16()
double	GetDouble()	SqlInt32	GetSqlInt32()
Type	GetFieldType()	SqlInt64	GetSqlInt64()
float	GetFloat()	SqlMoney	GetSqlMoney()
Guid	GetGuid()	SqlSingle	GetSqlSingle()
short	GetInt16()	SqlString	GetSqlString()
int	GetInt32()	object	GetSqlValue()
long	GetInt64()	int	GetSqlValues()
string	GetName()	string	GetString()
int	GetOrdinal()	object	GetValue()
DataTable	GetSchemaTable()	int	GetValues()
bool	Read()	bool	IsDBNull()
		bool	NextResult()

Nguyễn Văn Phúc - Khoa Công nghệ Thông tin - Đại học Đà Lạt

5

## Lớp DataReader

- Một cột trong DataTable có các thông tin sau
  - ColumnName** Tên cột
  - ColumnOrdinal** Thứ tự của cột
  - ColumnSize** Kích thước tối đa của dữ liệu
  - NumericPrecision** Số lượng chữ số
  - NumericScale** Số chữ số bên phải dấu chấm
  - ...
- Nên sử dụng các phương thức có dạng **GetSql\***
  - Cải thiện hiệu suất
  - Bảo đảm tính đúng của dữ liệu
  - Namespace: System.Data.SqlTypes

Nguyễn Văn Phúc - Khoa Công nghệ Thông tin - Đại học Đà Lạt

9

## Tạo đối tượng SqlDataReader

- Đối tượng SqlDataReader chỉ có thể được tạo bằng cách gọi phương thức ExecuteReader của đối tượng Command.

- Ví dụ

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

mySqlCommand.CommandText = "SELECT TOP 5 ProductID, ProductName, UnitPrice,
    UnitsInStock, Discontinued FROM Products ORDER BY ProductID";

mySqlConnection.Open();

SqlDataReader productsSqlDataReader = mySqlCommand.ExecuteReader();
```

Nguyễn Văn Phúc - Khoa Công nghệ Thông tin - Đại học Đà Lạt

7

## Đọc dữ liệu từ DataReader

- Sử dụng phương thức Read để đọc một dòng
  - True Cho biết còn dòng khác để đọc
  - False Nghĩa là tất cả các dòng đều đã được đọc
- Lấy giá trị của một cột theo cú pháp
  - myDataReader [ Chỉ\_Số\_Cột ]
  - myDataReader [ "Tên\_Cột" ]

- Ví dụ

```
while (productsSqlDataReader.Read())
{
    Console.WriteLine (productsSqlDataReader [ 0 ] );
    Console.WriteLine (productsSqlDataReader [ "ProductName" ] );
}
```

Nguyễn Văn Phúc - Khoa Công nghệ Thông tin - Đại học Đà Lạt

8

## Đọc dữ liệu từ DataReader

- Ưu - nhược điểm
  - Dùng chỉ số
  - Dùng tên cột
- Giải pháp
  - Sử dụng phương thức `GetOrdinal()`
  - Đây là cách hiệu quả nhất: linh động, mềm dẻo
- Ví dụ:
 

```
int productIDColPos = productsSqlDataReader.GetOrdinal( "ProductID" );
Console.WriteLine (productsSqlDataReader [ productIDColPos ] );
```

## Đóng kết nối

- Chỉ được đóng và phải đóng kết nối sau khi đọc dữ liệu
  - Vì DataReader sử dụng đối tượng Connection
  - Nếu không đóng, kết nối vẫn được mở và dành riêng cho DataReader trong khi không còn lệnh nào được thực hiện
- Đóng kết nối
 

```
mySqlConnection. Close ()
```
- Nếu có dùng `CommandBehavior.CloseConnection`

```
productsSqlDataReader. Close();
```

## Lấy dữ liệu với kiểu cụ thể

- Cú pháp: `C# Type variable = myDataReader. Get* (thứ_tự_cột)`  
`C# Type variable = myDataReader. Get* ("tên_cột")`

SQL SERVER TYPE	STANDARD C# TYPE	GET* METHOD
bigint	long	GetInt64()
bit	bool	GetBoolean()
char	string	GetString()
datetime	DateTime	GetDateTime()
decimal	decimal	GetDecimal()
float	double	GetDouble()
image	byte[]	GetBytes()
int	int	GetInt32()
money	decimal	GetDecimal()
nchar	string	GetString()
ntext	string	GetString()
nvarchar	string	GetString()
real	float	GetFloat()
smallint	short	GetInt16()
smallmoney	decimal	GetDecimal()
text	string	GetString()
tinyint	byte	GetByte()
varchar	string	GetString()

## Lấy dữ liệu với kiểu cụ thể

```
while (productsSqlDataReader.Read())
{
    int productID = productsSqlDataReader.GetInt32(productIDColPos);
    Console.WriteLine("productID = " + productID);

    string productName = productsSqlDataReader.GetString(productNameColPos);
    Console.WriteLine("productName = " + productName);

    decimal unitPrice = productsSqlDataReader.GetDecimal(unitPriceColPos);
    Console.WriteLine("unitPrice = " + unitPrice);

    short unitsInStock = productsSqlDataReader.GetInt16(unitsInStockColPos);
    Console.WriteLine("unitsInStock = " + unitsInStock);

    bool discontinued = productsSqlDataReader.GetBoolean(discontinuedColPos);
    Console.WriteLine("discontinued = " + discontinued);
}
```

## Lấy kiểu dữ liệu của một cột

- Sử dụng một trong hai phương thức
  - `GetFieldType` lấy tên kiểu dữ liệu chuẩn C#
  - `GetDataTypeName` lấy tên kiểu dữ liệu SQL Server.
- Ví dụ

```
Console.WriteLine("ProductID .NET type = " +
    productsSqlDataReader.GetFieldType ( productIDColPos ));

// Kết quả
ProductID .NET type = System.Int32

Console.WriteLine("ProductID database type = " +
    productsSqlDataReader.GetDataTypeName ( productIDColPos ));

// Kết quả
ProductID database type = int
```

## Lấy dữ liệu bởi phương thức GetSql\*

SQL SERVER TYPE	Sql* TYPE	GetSql* METHOD
bigint	SqlInt64	GetSqlInt64()
int	SqlInt32	GetSqlInt32()
smallint	SqlInt16	GetSqlInt16()
tinyint	SqlByte	GetSqlByte()
bit	SqlBoolean	GetSqlBoolean()
decimal	SqlDecimal	GetSqlDecimal()
numeric	SqlDecimal	GetSqlDecimal()
money	SqlMoney	GetSqlMoney()
smallmoney	SqlMoney	GetSqlMoney()
float	SqlDouble	GetSqlDouble()
real	SqlSingle	GetSqlSingle()
datetime	SqlDateTime	GetSqlDateTime()
char	SqlString	GetSqlString()
varchar	SqlString	GetSqlString()
text	SqlString	GetSqlString()
nchar	SqlString	GetSqlString()
nvarchar	SqlString	GetSqlString()
ntext	SqlString	GetSqlString()

## Lấy dữ liệu bởi phương thức GetSql\*

```
while (productsSqlDataReader. Read ())
{
    SqlInt32 productID = productsSqlDataReader. GetSqlInt32 (productIDColPos);
    Console.WriteLine("productID = " + productID);

    SqlString productName = productsSqlDataReader. GetSqlString (productNameColPos);
    Console.WriteLine("productName = " + productName);

    SqlMoney unitPrice = productsSqlDataReader. GetSqlMoney (unitPriceColPos);
    Console.WriteLine("unitPrice = " + unitPrice);

    SqlInt16 unitsInStock = productsSqlDataReader. GetSqlInt16 (unitsInStockColPos);
    Console.WriteLine("unitsInStock = " + unitsInStock);

    SqlBoolean discontinued =
        productsSqlDataReader. GetSqlBoolean (discontinuedColPos);
    Console.WriteLine("discontinued = " + discontinued);
}
```

## Xử lý giá trị NULL

- Kiểm tra một cột có chứa giá trị **NULL**

- Dùng phương thức **IsDBNull**
  - **True** Cột có chứa giá trị null
  - **False** Không cho phép chứa giá trị null

```
if (productsSqlDataReader.IsDBNull(unitPriceColPos))
{
    Console.WriteLine("UnitPrice column contains a null value");
}
else
{
    unitPrice = productsSqlDataReader.GetDecimal (unitPriceColPos);
}
```

- Nhận xét

- Giá trị **null** cho biết giá trị của cột đó là chưa biết.
- Các kiểu dữ liệu chuẩn của C# không thể biểu diễn (hay lưu trữ) giá trị **null**.
  - Vì thế, ta phải dùng kiểu **Sql\***.



## Xử lý giá trị NULL

```
SqlMoney unitPrice = productsSqlDataReader. GetSqlMoney (unitPriceColPos);
Console.WriteLine("unitPrice = " + unitPrice);
```

- Kết quả xuất: `unitPrice = Null`

```
Console.WriteLine("unitPrice.IsNull = " + unitPrice.IsNull);
```

- Kết quả xuất: `unitPrice.IsNull = true`

## Thực thi nhiều lệnh truy vấn SQL

```
SqlCommand mySqlCommand = mySqlConnection.CreateCommand();
mySqlCommand.CommandText =
    "SELECT TOP 5 ProductID, ProductName FROM Products ORDER BY ProductID;" +
    "SELECT TOP 3 CustomerID, CompanyName FROM Customers ORDER BY CustomerID;" +
    "SELECT TOP 6 OrderID, CustomerID FROM Orders ORDER BY OrderID;";

SqlDataReader productsSqlDataReader = mySqlCommand. ExecuteReader();
do
{
    while (mySqlDataReader. Read())
    {
        Console.WriteLine("mySqlDataReader[0] = " + mySqlDataReader [0] );
        Console.WriteLine("mySqlDataReader[1] = " + mySqlDataReader [1] );
    }
} while (mySqlDataReader. NextResult());
```

## Thực hành

- Làm việc theo nhóm
- Tiếp tục thực hiện các yêu cầu theo từng đề tài
  - Áp dụng kiến thức vừa được học để viết các hàm theo yêu cầu
  - Viết báo cáo