

# Chương 1: Giới thiệu chung về hệ quản trị cơ sở dữ liệu

1.1 Thế nào là một cơ sở dữ liệu?

1.2 Thế nào là một hệ quản trị cơ sở dữ liệu?

1.2.1 Định nghĩa: Một HQTCSDDL là một tập hợp chương trình giúp cho người sử dụng tạo ra, duy trì và khai thác một cơ sở dữ liệu: Nó là một hệ thống phần mềm phổ dụng, làm dễ quá trình định nghĩa, xây dựng và thao tác cơ sở dữ liệu cho các ứng dụng khác nhau

# Chương 1: Giới thiệu chung về hệ quản trị cơ sở dữ liệu

## 1.2.2 Một số chức năng quan trọng của HQTCSDL

1. Lưu trữ các định nghĩa, các mối liên kết dữ liệu. Các chương trình truy cập đến cơ sở dữ liệu làm việc thông qua HQTCSDL
1. Tạo ra một hệ thống bảo mật và áp đặt tính bảo mật và riêng tư trong cơ sở dữ liệu.
2. Tạo ra các cấu trúc phức tạp cho phép nhiều người sử dụng truy cập đến cơ sở dữ liệu.
3. Cung cấp các thủ tục sao lưu và phục hồi dữ liệu để đảm bảo sự an toàn và toàn vẹn dữ liệu.
4. Cung cấp việc truy cập dữ liệu thông qua một ngôn ngữ truy vấn.

# Chương 1: Giới thiệu chung về hệ quản trị cơ sở dữ liệu

## 1.3 Giới thiệu một số loại HQTCSDL

Access, Foxpro, DB2, Sysbase (Sysbase System), Oracle, MySQL, SQL Server (6.0, 6.5, 7.0, dòng 2000)

## 1.4 Ngôn ngữ HQTCSDL

## 1.5. Ví dụ

## 1.6 Tổng kết chương: Sinh viên hiểu thế nào là một CSDL và thế nào là một HQTCSDL?

# Chương 2: Giới thiệu hệ quản trị cơ sở dữ liệu SQL Server

## 2.1 Giới thiệu về SQL Server

### 2.1.1 Giới thiệu SQL Server

- Là sản phẩm về phương tiện cơ sở dữ liệu hàng đầu của Microsoft.
- Chiếm thị phần nhiều nhất.
- Dùng được cho các hệ điều hành Window NT 4.0 (từ Service Pack 5 trở đi), Dòng Windows 2000, Window 98 hay ME.
- Tham khảo <http://www.microsoft.com/sql/>.

# Chương 2: Giới thiệu hệ quản trị cơ sở dữ liệu SQL Server

## 2.1.2 Lịch sử của SQL Server

- Microsoft cùng với Sysbase phát triển phiên bản SQL Server đầu tiên sử dụng trên hệ điều hành IBM OS/2.
- SQL Server phiên bản 4.2 sau đó là 4.2.1 được phát triển cho hệ điều hành Window NT.
- Sysbase tách khỏi Microsoft phát triển các hệ Sysbase (mới nhất là Adaptive Server Enterprise).
- Microsoft phát triển tiếp các hệ SQL Server 6.0, 6.5 chạy trên Window NT. SQL Server phiên bản 7.0 chạy trên các hệ điều hành dòng Windows 2000, Window 98 hay ME. Sau này là các dòng SQL Server 2000.

# Chương 2: Giới thiệu hệ quản trị cơ sở dữ liệu SQL Server

## 2.1.3 Các ấn bản và yêu cầu phần cứng

- Personal Edition
- Standard Edition
- Enterprise Edition

Yêu cầu phần cứng: Tối thiểu Pentium II, 32 MB

# Chương 2: Giới thiệu hệ quản trị cơ sở dữ liệu SQL Server

## 2.2 Cài đặt và khởi động với SQL Server

Ví dụ về cài đặt và các thiết lập tùy chọn trong quá trình cài đặt

2.3 Tổng kết chương: Sinh viên hiểu sơ lược về hệ quản trị SQL Server, biết cách cài đặt với những tùy chọn.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.1 Các đối tượng và thuộc tính của một cơ sở dữ liệu

- Table (bảng).
- Field (trường) hay Column (cột): Phải được gán một kiểu dữ liệu và phải có tên duy nhất.
- Data Type (Kiểu dữ liệu): Có nhiều kiểu dữ liệu khác nhau, mỗi một kiểu được gán cho một trường nào đó bên trong bảng.
- Primary Key (Khóa chính): Là duy nhất trong các hàng, cung cấp cách xác định duy nhất một bản ghi.
- Foreign Key (Khóa ngoại): Là các cột tham chiếu đến khóa chính hay các ràng buộc duy nhất của một bảng khác.
- Constrains (Ràng buộc): Ràng buộc là cơ chế nhằm đảm bảo tính toàn vẹn dữ liệu, phụ thuộc vào hệ thống và Server.



# Chương 3: Quản trị CSDL trong SQL Server

- Defaults (Mặc định): Một mặc định có thể xác lập trên một trường sao cho nếu không có dữ liệu nào được đưa vào thì giá trị mặc định sẽ được dùng đến.
- View (Khung nhìn hay bảng ảo): Về cơ bản là các truy vấn lưu trữ trong cơ sở dữ liệu để có thể tham chiếu tới một hay nhiều bảng. Có thể cho phép nhìn thấy hay không nhìn thấy một số cột trong một hay nhiều bảng nào đó. Cũng có thể dùng View làm kỹ thuật an toàn.
- Stored procedure (Thủ tục lưu trữ): Giống như các Macro, trong đó mã lệnh SQL có thể được viết và lưu trữ mang một tên.
- Triggers (Các bẫy lỗi – còn dịch là bộ kích khởi): Là các thủ tục lưu trữ kích hoạt trước hoặc sau khi bổ xung, sửa chữa hoặc xóa dữ liệu ra khỏi cơ sở dữ liệu.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2 Khái niệm về quản trị và trình quản trị SQL Server Enterprise Manager

### a. Các cách khởi động:

- Start\Program\Microsoft SQL Server\Enterprise Manager.
- Chạy Start\Run\mmc.exe khởi động Console, từ trình đơn File chọn SQL Server Enterprise Manager.MSC.
- Khi quản trị nếu chưa có máy chủ (mặc định là (LOCAL) (Windows NT)) thì cần phải tạo một máy chủ bằng cách: bấm chuột phải vào SQL Server Group chọn New SQL Server Registration, sau đó nhập tên Server và chọn chế độ xác thực. Có thể dùng nhiều máy chủ, tùy từng phiên bản của SQL Server mà có thể đăng ký được bao nhiêu máy chủ.

### b. Chức năng: Quản trị cơ sở dữ liệu bằng giao diện đồ họa.

### c. Ví dụ

# Chương 3: Quản trị CSDL trong SQL Server

## 3.3 Giới thiệu SQL Query Analyzer

### a. Khởi động:

- Ba cách khởi động thứ nhất chọn Tool\SQL Query Analyzer trong SQL Server Enterprise
- Thứ hai Start\Program\Microsoft SQL Server\Query Analyzer.
- Thứ ba chọn Start\Run\isqlw.exe
- Trong cả ba cách chọn thì trình SQL Query Analyzer đều yêu cầu một kết nối tới máy chủ SQL Server.

### b. Chức năng:

- Là giao diện chính cho việc chạy các truy vấn SQL và các thủ tục lưu trữ
- Quản trị cơ sở dữ liệu bằng các lệnh SQL, còn SQL Server Enterprise Manager quản trị bằng giao diện.

### c. Ví dụ

# Chương 3: Quản trị CSDL trong SQL Server

## 3.4 Ví dụ về tạo, mở, đóng, đổi tên, xóa một cơ sở dữ liệu

a. **Tạo:** Hai cách hoặc dùng SQL Server Enterprise Manager bằng cách bấm chuột phải vào Databases chọn New Database...

Hoặc dùng cửa sổ SQL Query Analyzer bằng cách gõ tập lệnh:

```
Use master
```

```
go
```

```
Create Database Vidu on Primary (Name = ViduData,
```

```
Filename = 'C:\Program Files\Microsoft SQL Server\MSSQL\Data\ViduData.mdf',
```

```
Size = 4MB,
```

```
Maxsize = 10MB,
```

```
Filegrowth = 2MB)
```

```
Log ON
```

```
(Name = ViduLog,
```

```
FileName='C:\Program Files\Microsoft SQL Server\MSSQL\Data\ViduLog.ldf',
```

```
size = 2MB,
```

```
maxsize = 6MB,
```

```
Filegrowth = 1MB)
```

```
Go
```

# Chương 3: Quản trị CSDL trong SQL Server

## b. Xóa:

- Hai cách thứ nhất dùng SQL Server Enterprise Manager bấm chuột phải vào CSDL cần xóa chọn delete
- Thứ hai dùng lệnh SQL trong SQL Query Analyzer

*Use master*

*Go*

*Drop Database name1, name2,...*

*Go*

## c. Muốn đổi tên dùng tập lệnh

*use Ten\_CSDL\_muon\_doi\_ten*

*Go*

*exec sp\_dboption Ten\_CSDL\_muon\_doi\_ten, 'Single User',true*

*exec sp\_renamedb 'ten\_cu', 'ten\_moi'*

*exec sp\_dboption ten\_moi, 'Single User', false*

*Go*

Trong SQL Query Analyzer.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2 Bảo mật và Quyền trong SQL Server

### 3.2.1 Mô hình đăng nhập vào SQL Server

Mô hình ba mức :



# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.2. Mức thứ nhất Sự chứng thực của hệ điều hành Windows (Windows 2000 Network Authentication)

- SQL Server được chứa trên một máy chủ cài hệ điều hành Windows, vì vậy trước tiên muốn truy cập đến SQL Server thì phải được phép truy cập vào máy chủ, tức là phải qua được bước chứng thực của hệ điều hành Windows cài trên máy chủ với một phép chứng thực hợp lệ nào đó.
- Ví dụ muốn vào nhà trước tiên phải đi qua cửa.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.3. Mức thứ hai là chứng thực đăng nhập vào SQL Server (SQL Server Login Authentication)

Có hai loại thứ nhất là

- Chế độ tích hợp Windows Authentication Mode (chỉ dùng trong hệ điều hành Windows 2000, NT), tức là dùng chính tài khoản đăng nhập vào Windows để đăng nhập vào SQL Server với sự “cho phép” của người quản trị SQL Server, lúc này SQL Server “nhờ” Windows chứng thực tài khoản hộ.
- Ví dụ: Dùng Computer Management\Local User and Groups tạo một tài khoản (hoặc nhóm), đăng nhập SQL Server với tư cách là nhà quản trị sau đó “cấp phép” cho tài khoản (hoặc nhóm) này được đăng nhập vào SQL Server dưới chế độ Windows Authentication.



# Chương 3: Quản trị CSDL trong SQL Server

“Cấp phép” bằng hai cách dùng giao diện hoặc bằng câu lệnh SQL:

- *Dùng giao diện*: bằng cách mở cửa sổ Enterprise Manager chọn thư mục Security, bấm chuột phải vào Login chọn new login, bấm chuột vào nút bấm có dấu “...” chọn tài khoản (hoặc nhóm) cần cấp phép và bấm nút Add, sau đó bấm OK, ở trường hợp này như trên ta chọn chế độ Windows Authentication, cuối cùng là bấm OK.
- *Dùng câu lệnh*: Gõ lệnh SQL trong cửa sổ SQL Query Analyzer như sau:

*Use Tên một cơ sở dữ liệu nào đó (mặc định là master)*

*Exec sp\_grantlogin ‘Domain(Server)\tài khoản’*

Trong đó ‘Domain(Server)\tài khoản’ có thể là một tài khoản hoặc nhóm.

# Chương 3: Quản trị CSDL trong SQL Server

- Để gỡ bỏ quyền trong giao diện đồ họa chỉ đơn giản là bấm chuột phải vào tài khoản (hoặc nhóm) cần xóa và chọn delete.
- Dùng câu lệnh ta gõ lệnh:

*Exec sp\_revokelogin 'Domain(Server)\tài khoản'*

*hoặc*

*Exec sp\_denylogin 'Domain(Server)\tài khoản'*

Lưu ý: nếu trong trường hợp 'Domain(Server)\tài khoản' là một nhóm trong Windows thì muốn gỡ bỏ quyền đăng nhập của một người trong nhóm đó nhất thiết phải dùng

*Exec sp\_denylogin 'Domain(Server)\tài khoản'* với

*'Domain(Server)\tài khoản'* là tài khoản của người cần gỡ bỏ.

# Chương 3: Quản trị CSDL trong SQL Server

Thứ hai là

- Chế độ hỗn hợp Mixed Mode bao gồm chế độ tích hợp trên cộng với chế độ chứng thực SQL Server, chế độ chứng thực SQL Server là chế độ trong đó SQL Server chấp nhận một ID đăng nhập và mật khẩu từ một người sử dụng, và kiểm tra các Ủy nhiệm hợp lệ mà không cần bất kỳ một sự giúp đỡ nào từ hệ điều hành Windows. Xảy ra trường hợp này khi trên mạng có những máy không cài hệ điều hành Windows (Unix chẳng hạn) muốn truy cập vào SQL Server.
- Ví dụ: Dùng Computer Management\Local User and Groups tạo một tài khoản, đăng nhập SQL Server với tư cách là nhà quản trị sau đó “cấp phép” cho tài khoản này được đăng nhập vào SQL Server với chế độ SQL Server Authentication với một password nào đó.

# Chương 3: Quản trị CSDL trong SQL Server

“Cấp phép” bằng giao diện thì như trên tuy nhiên ở trường hợp này ta chọn chế độ SQL Server Authentication với một password nào đó sau đó cuối cùng là bấm OK.

Dùng câu lệnh thì gõ:

*Use Tên một cơ sở dữ liệu nào đó (mặc định là master)*

*Exec sp\_addlogin ‘username’, ‘password’*

Để gỡ bỏ dùng giao diện đồ họa hoặc dùng câu lệnh:

*Exec sp\_droplogin ‘username’*

Thay đổi password có thể dùng giao diện đồ họa hoặc câu lệnh

*Exec sp\_password ‘old’, ‘new’, ‘username’*

Tất nhiên để làm được tất cả các điều này người cấp phép phải đăng nhập SQL Server với tư cách là người quản trị.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.4. Yêu cầu chứng thực như một người dùng cơ sở dữ liệu (SQL Server Database User Authentication)

### a. Người dùng cơ sở dữ liệu

- Việc có tài khoản truy cập được vào SQL Server không có nghĩa là đã có quyền truy cập vào được một cơ sở dữ liệu cụ thể nào đó. Muốn làm việc được với một cơ sở dữ liệu cụ thể nào đó bạn phải có một tài khoản trong cơ sở dữ liệu đó, tài khoản này nằm trong bảng sysusers của cơ sở dữ liệu đó. Tài khoản này ta còn gọi là người dùng CSDL.
- Mỗi một CSDL tạo ra thì luôn có một người dùng mặc định là chủ của cơ sở dữ liệu đó (dbo), cũng chính là người đã tạo ra CSDL đó, người dùng này hiển nhiên là có toàn quyền đối với CSDL đó.
- Các CSDL mặc định của SQL Server (như master, pubs,...) còn có một người dùng mặc định khác là Guest.

# Chương 3: Quản trị CSDL trong SQL Server

- Để tạo một người dùng trong một CSDL cụ thể thông thường ta dùng chính tài khoản đã đăng nhập vào SQL Server luôn (có thể dùng một tên khác) để dễ dàng cho việc nhớ và quản lý.

## b. Tạo người dùng trong CSDL

Hai cách

- Thứ nhất dùng giao diện: Chọn CSDL cụ thể mà ta muốn tạo một người dùng, bấm chuột phải vào Users sau đó chọn New Database User, trong mục login name chọn tài khoản mà ta cho phép dùng CSDL này với user name mặc định ta chọn trùng với login name cho dễ quản lý, cuối cùng là bấm OK.

# Chương 3: Quản trị CSDL trong SQL Server

- Thứ hai dùng câu lệnh:

*Use Tên CSDL*

*Exec sp\_grantdbaccess 'Domain(Server)\tài khoản', 'tên người dùng'*

*Ví dụ:*

*Exec sp\_grantdbaccess 'TRINHCUONG\test'*

## **c. GỖ BỎ**

- Dùng giao diện chỉ đơn giản là bấm chuột phải vào người dùng đó và chọn delete.
- Dùng câu lệnh thì gõ lệnh:

*Exec sp\_revokedbaccess 'tên người dùng'*

*Ví dụ: Exec sp\_revokedbaccess 'TRINHCUONG\test'*

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.5 Các vai trò (Role)

- Vai trò có nghĩa là tập hợp các quyền được làm một việc gì đấy trong CSDL, có hai mức là vai trò ở mức hệ quản trị SQL Server (ta hay gọi là mức Server) và mức CSDL (mức CSDL).

### 3.2.5.1 Vai trò ở mức hệ quản trị SQL Server (Serverwide)

#### a. Định nghĩa các vai trò Serverwide

- Có tám vai trò, không thể thêm hay xóa các vai trò này, mỗi người dùng khi có tài khoản đăng nhập vào SQL Server đều có thể được gán một trong những vai trò này (ta có thể hiểu là các tài khoản đăng nhập có thể được gán một quyền nào đó).



# Chương 3: Quản trị CSDL trong SQL Server

- Sysadmin: Thành viên của vai trò này có thể làm bất cứ điều gì trong SQL Server. Chúng xuất hiện là dbo của mỗi CSDL. Về cơ bản chúng ghi đè lên các hệ thống quyền hạn và bảo mật.
- Serveradmin: Thành viên của vai trò này có thể thiết lập các tùy chọn cấu hình và có thể ngừng hoạt động máy chủ.
- Setupadmin: Thành viên của vai trò này có thể cài đặt và định cấu hình liên kết với các máy chủ và đánh dấu một thủ tục để chạy khi khởi động.
- Securityadmin: Thành viên của vai trò này có thể tạo và điều khiển các đăng nhập của máy chủ cũng như các quyền hạn để tạo các CSDL. Có thể xác lập lại các mật khẩu của các đăng nhập SQL Server (ngoại trừ các thành viên của quyền sysadmin).

# Chương 3: Quản trị CSDL trong SQL Server

- Processadmin: Thành viên của vai trò này có thể điều khiển quá trình chạy trên CSDL của máy chủ, có thể hủy các truy vấn và hướng dẫn người dùng những việc phải làm khi thực hiện các truy vấn.
- Dbcreator: Thành viên của vai trò này có thể tạo, xóa và sửa các CSDL trên máy chủ, họ cũng có thể khôi phục bất kỳ cơ sở dữ liệu nào trên máy chủ.
- Bulkadmin: Thành viên của vai trò này có thể chạy phát biểu BULK INSERT.
- Diskadmin: Thành viên của vai trò này có thể quản lý các tập tin và sự tăng trưởng của các tập tin, tuy nhiên thường dùng trong SQL Server 6.5 (nên ít được quan tâm ở đây).

# Chương 3: Quản trị CSDL trong SQL Server

## b. Gán một đăng nhập vào một Server Role

Có hai cách dùng giao diện hoặc dòng lệnh

- Thứ nhất dùng giao diện: Chọn Login trong Security, sau đó bấm chuột phải vào đăng nhập cần gán chọn Properties, chọn mục Server Roles, sau đó tích vào những vai trò mà mình muốn cấp quyền cho đăng nhập. Ta cũng có thể làm tương tự như vậy khi tạo mới một đăng nhập.
- Thứ hai dùng dòng lệnh thì ta gõ lệnh:

*Exec sp\_addsrvrolemember 'login', 'role'*

Ví dụ: *Exec sp\_addsrvrolemember 'TRINHCUONG\test', 'dbcreator'*

# Chương 3: Quản trị CSDL trong SQL Server

## c. Gỡ bỏ

- Dùng giao diện thì chỉ đơn giản là lặp lại bước trên và bỏ dấu tích vào vai trò mà ta muốn gỡ bỏ.
- Dùng dòng lệnh thì gõ lệnh:

*Exec sp\_dropsvrolemember 'login', 'role'*

*Ví dụ: Exec sp\_dropsvrolemember 'TRINHCUONG\test', 'dbcreator'*

### 3.2.5.2 Vai trò ở mức CSDL

- Mỗi CSDL đều chứa các vai trò, có khoảng 9 vai trò là cố định mà bất kỳ CSDL nào cũng đều có, không thể xóa bất kỳ một vai trò nào trong số này, một số vai trò trong đó là có thể sửa được. Ngoài ra người dùng cũng có thể tự định nghĩa các vai trò cho riêng mình. Mỗi vai trò chỉ ảnh hưởng trong một CSDL cụ thể.

# Chương 3: Quản trị CSDL trong SQL Server

## a. Các vai trò CSDL cố định

- **Db\_owner** : Thành viên của vai trò này có thể thực hiện bất kỳ điều gì họ muốn, tất nhiên là chỉ ở CSDL mà ta đang xét đến, quyền giống hệt như người dùng dbo ngoại trừ việc không thực hiện được câu lệnh khôi phục CSDL.
- **Db\_accessadmin** : Thành viên của vai trò này có thể thêm hoặc gỡ bỏ quyền truy cập vào CSDL của người dùng (ví dụ chạy thủ tục `sp_grantdbaccess`).
- **Db\_securityadmin** : Thành viên của vai trò này có thể điều khiển các quyền hạn, các vai trò, các thành viên của vai trò và các chủ nhân của các đối tượng trong CSDL.
- **Db\_ddladmin** : Thành viên của vai trò này có thể tạo, sửa đổi và xóa tất cả các đối tượng của CSDL nhưng họ không thể đưa ra các câu lệnh liên quan đến bảo mật (`grant`, `revoke`, `deny`).

# Chương 3: Quản trị CSDL trong SQL Server

- Db\_backupoperator : Thành viên của vai trò này có quyền đưa ra các điểm kiểm tra cũng như các câu lệnh sao lưu.
- Db\_datareader : Thành viên của vai trò này có quyền select trên bất kỳ bảng, bảng nhìn, hoặc hàm nào trong CSDL.
- Db\_datawriter : Thành viên của vai trò này có các quyền chèn, cập nhật, và xóa trên bất kỳ bảng, bảng nhìn trong CSDL.
- Db\_denydatareader : Thành viên của vai trò không thể select trên bất kỳ bảng, bảng nhìn hoặc hàm nào trong CSDL.
- Db\_denydatawriter : Thành viên của vai trò không thể sửa đổi bất kỳ dữ liệu nào trong CSDL bằng các phát biểu insert, update, hoặc delete trên bất kỳ bảng hoặc bảng nhìn nào.

# Chương 3: Quản trị CSDL trong SQL Server

## b. Các vai trò CSDL do người dùng tự định nghĩa

- Người dùng có thể thêm các vai trò do mình tự định nghĩa, các vai trò có thể chứa các vai trò khác là thành viên, và tất nhiên một user có thể là thành viên của bất kỳ vai trò nào.

Để tạo một vai trò có hai cách

- Thứ nhất dùng giao diện: Chọn CSDL cần tạo vai trò, bấm chuột phải vào Roles, chọn New Database Role, đặt tên vai trò mà bạn muốn vào hộp name, trong mục Add bạn có thể thêm vào các thành viên của vai trò mà bạn muốn, thành viên có thể là người dùng cũng có thể là các vai trò khác, để gỡ bỏ thành viên của vai trò chỉ đơn giản bạn chọn thành viên cần gỡ bỏ và bấm remove, cuối cùng bấm OK. Để xóa một vai trò bạn phải xóa tất cả các thành viên của vai trò sau đó bấm chuột phải vào vai trò cần xóa và chọn delete.

# Chương 3: Quản trị CSDL trong SQL Server

- Thứ hai dùng câu lệnh: Để tạo vai trò mới dùng câu lệnh:

*Use Tên CSDL cần tạo vai trò*

*Exec sp\_addrole ‘tên vai trò’, ‘tên người dùng là chủ vai trò (nếu không có thì mặc định là dbo)’*

Tên vai trò phải là duy nhất trong một CSDL

Để gỡ bỏ dùng câu lệnh:

*Use Tên CSDL cần gỡ bỏ vai trò*

*Exec sp\_droprole ‘tên vai trò’*

Để thêm một thành viên vào vai trò dùng câu lệnh:

*Use Tên CSDL*

*Exec sp\_addrolemember ‘tên vai trò’, ‘database\_account’*

Trong đó database\_account là tên người dùng, tên nhóm hoặc tên vai trò mà bạn muốn thêm vào vai trò này.



# Chương 3: Quản trị CSDL trong SQL Server

Để gỡ bỏ một thành viên từ một vai trò dùng câu lệnh:

*Use Tên CSDL*

*Exec sp\_droprolemember 'tên vai trò', 'database\_account'*

Database\_account giống như trên.

- Ví dụ: Mã lệnh sau trình bày cách thêm vai trò mới trong CSDL pubs, và rồi gán người dùng, vai trò vào nó:

*Use Pubs*

*Exec sp\_addrole '5b'*

*Exec sp\_addrole '5c'*

*Exec sp\_addrolemember '5b', 'TRINHCUONG\test'*

*Exec sp\_addrolemember '5c', '5b'*

*Exec sp\_addrolemember '5c', 'Bob'*

# Chương 3: Quản trị CSDL trong SQL Server

- Ở đây TRINHCUONG\test và Bob đều là tên người dùng CSDL, TRINHCUONG\test là trường hợp mà tên đăng nhập trùng luôn với tên người dùng CSDL (thông thường là như vậy). Lưu ý nếu tài khoản đăng nhập có tên domain (hoặc Server) đằng trước\ tên đăng nhập đằng sau ví dụ như dạng TRINHCUONG\test thì đây là dạng đăng nhập Windows Authentication.
- Để xóa Vai trò thì trước hết vai trò phải không chứa thành viên nào. Muốn xóa vai trò 5c ta phải thực hiện như sau:

*Use Pubs*

*Exec sp\_droprolemember '5c', '5b'*

*Exec sp\_droprolemember '5c', 'Bob'*

*Exec sp\_droprole '5c'*

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.6 Thiết lập việc cấp quyền truy cập CSDL

- Trong các phần trước chúng ta đã biết nếu có một đăng nhập vào SQL Server thì cũng chưa nói lên điều gì về khả năng làm việc với một CSDL cụ thể, thậm chí khi có một tài khoản người dùng trong một CSDL cụ thể thì cũng không có nghĩa là đã được quyền làm mọi việc với CSDL đó, một tài khoản người dùng CSDL có khả năng làm được gì trong CSDL phải qua bước cấp quyền.

# Chương 3: Quản trị CSDL trong SQL Server

- Cấp quyền là một cơ chế bảo mật tốt của SQL Server, giúp cho người quản trị quản lý tốt được CSDL, phòng chống được những hư hại có thể xảy ra với CSDL hơn là phải đi tìm cách khắc phục những hư hại có thể xảy ra, người quản trị có thể khống chế được người dùng CSDL bằng cách cấp quyền cho phép họ có thể làm được và không làm được việc gì trong một CSDL cụ thể.
- Có hai loại cấp quyền truy cập CSDL thứ nhất là Quyền phát biểu (Statement Permissions) và thứ hai là Quyền về đối tượng (Object Permissions).

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.6.1 Thiết lập việc cấp các quyền phát biểu

Có hai cách:

### a. Thứ nhất là dùng giao diện

- Bấm chuột phải vào CSDL mà ta muốn cấp quyền chọn mục Properties, chọn tiếp mục Permissions, hiện ra một bảng danh sách các quyền và danh sách các người dùng và vai trò, để cấp quyền ta chỉ đơn giản là tích vào để hiện dấu tích màu xanh, để vô hiệu (deny) một quyền ta tích làm cho dấu đỏ xuất hiện, để gỡ bỏ ta để trống ô đấy. Lưu ý rằng quyền tạo CSDL (create database) chỉ có trong CSDL master.

# Chương 3: Quản trị CSDL trong SQL Server

## b. Thứ hai là dùng câu lệnh

- Cấp quyền:

*Use Tên CSDL*

*Grant All hoặc danh sách các quyền To Account*

Trong đó Account có thể là tên người dùng CSDL hoặc vai trò.

Ví dụ: *Grant Create table, create view to role1, role2*

Câu lệnh trên gán quyền tạo bảng và tạo view cho vai trò role1 và role2

*Grant All to [TRINHCUONG\test]*

Câu lệnh này gán tất cả các quyền trong một CSDL cho TRINHCUONG\test

# Chương 3: Quản trị CSDL trong SQL Server

- Gỡ bỏ:

*Use Tên CSDL*

*Revoke All hoặc danh sách các câu lệnh To Account*

Trong đó Account tương tự như trên

Ví dụ: *Revoke Create table, create view to role1, role2*

Câu lệnh này gỡ bỏ quyền tạo bảng và tạo view của role1 và role2

*Revoke All to [TRINHCUONG\test]*

Câu lệnh này gỡ bỏ tất cả các quyền của TRINHCUONG\test trong CSDL

# Chương 3: Quản trị CSDL trong SQL Server

- Vô hiệu:

*Use Tên CSDL*

*Deny All hoặc danh sách các câu lệnh To Account*

Trong đó Account tương tự như trên

Ví dụ: *deny Create table, create view to role1, role2*

Giả sử ta có TRINHCUONG\test là một thành viên của vai trò Role3

Câu lệnh: *Grant create table, create view to Role3*

Lúc này hiển nhiên là TRINHCUONG\test có hai quyền create table và create view, nếu ta không muốn TRINHCUONG\test có quyền tạo bảng ta dùng câu lệnh

*deny create table to TRINHCUONG\test*



# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.6.2 Thiết lập việc cấp các quyền về đối tượng

### a. Người sở hữu CSDL (dbo)

- Người dùng dbo có tất cả các quyền của các thành viên của vai trò db\_owner. Mỗi CSDL chỉ có một người dbo
- Được xem là người dbo của CSDL nếu bạn là người tạo lập CSDL, người đăng nhập để tạo CSDL là dbo, hoặc bạn được giao nhiệm vụ sở hữu CSDL thông qua thủ tục lưu trữ hệ thống sp\_changedbowner, hoặc bạn kết nối với SQL Server như là thành viên điều hành hệ thống (sysadmin), hay tài khoản sa.

# Chương 3: Quản trị CSDL trong SQL Server

## b. Người sở hữu đối tượng (dbo)

- Mặc nhiên người dùng nào tạo ra đối tượng CSDL thì là dbo của đối tượng đó, các thành viên của db\_owner và db\_ddladmin có thể tạo ra các đối tượng và định rõ tên người sở hữu đối tượng. Nếu bạn đăng nhập với quyền sysadmin hoặc dbowner khi tạo đối tượng thì mặc nhiên người sở hữu đối tượng là dbo.
- Để thuận tiện cho việc sử dụng và cấp quyền trong CSDL tốt nhất nên để các đối tượng được sở hữu duy nhất bởi người sở hữu đối tượng dbo, lý do là vấn đề liên hệ sở hữu đơn và liên hệ sở hữu đứt.

# Chương 3: Quản trị CSDL trong SQL Server

## c. Gán quyền về đối tượng

Có hai cách dùng giao diện hoặc câu lệnh

- Dùng giao diện: Mở CSDL mà bạn muốn xem hoặc hiệu chỉnh, nhấp chuột phải vào đối tượng mà bạn muốn hiệu chỉnh và chọn mục Properties, sau đó kích vào nút Permissions, lúc này bạn có thể cấp quyền cho một người dùng nào đó sử dụng một quyền nào đó (select, update, ...) bằng cách tích dấu xanh vào ô tương ứng, gỡ bỏ hay vô hiệu hoàn toàn tương tự như cách trước đây. Nếu bạn muốn cấp quyền ở mức cột thì kích vào nút columns và tích vào cột mà bạn muốn cấp quyền, cuối cùng chọn OK.

# Chương 3: Quản trị CSDL trong SQL Server

- Dùng dòng lệnh:
  - Lệnh cấp quyền Grant cho phép bất kì ai một hoặc nhiều quyền về đối tượng. Nó cũng đề lên quyền vô hiệu hóa:

*Grant All hoặc Tập lệnh [danh sách cột nếu có] On Đối tượng To Account [With Grant Option].*

*Ví dụ: Grant select on Chucdanh (tenchucdanhvt, tenchucdanhdd) to TRINHCUONG\test*

*Grant select (tenchucdanhvt, tenchucdanhdd) on Chucdanh to TRINHCUONG\test*

*Grant select, insert on Chucdanh to TRINHCUONG\test with grant option.*

# Chương 3: Quản trị CSDL trong SQL Server

- Lệnh Gỡ bỏ Revoke sẽ thu hồi một hay nhiều quyền được gán trước đây:

*Revoke All hoặc Tập lệnh [danh sách cột nếu có] On Đối tượng From Account [Cascade].*

Ví dụ: *Revoke select (tenchucdanhvvt, tenchucdanhdd) on Chucdanh from TRINHCUONG\test Cascade.*

- Lệnh Vô hiệu Deny sẽ vô hiệu hóa những quyền của một đối tượng cụ thể

*Deny All hoặc Tập lệnh [danh sách cột nếu có] On Đối tượng To Account [Cascade].*

Ví dụ: *Deny select, insert on chucdanh to TRINHCUONG\test*

# Chương 3: Quản trị CSDL trong SQL Server

## 3.2.6.3 Phát sinh kịch bản Script

- Dùng giao diện, kích chuột phải vào CSDL, hoặc đối tượng cần phát sinh kịch bản chọn nút All Tasks , chọn Generate SQL Script, chọn Show All .....

# Chương 3: Quản trị CSDL trong SQL Server

## 3.3 Bài tập

**Bài 1:** Cơ chế bảo mật trong SQL Server có mấy mức? Hãy trình bày tổng quan từng mức?

**Bài 2:** Sự khác nhau giữa chế độ đăng nhập một tài khoản (integrated) và chế độ hai tài khoản (SQL Server – authentication)? Khi nào phải dùng chế độ hai tài khoản?

**Bài 3:** Tôi có một tài khoản người dùng trong Windows, tôi đã được quyền đăng nhập vào SQL Server chưa? để có thể sử dụng được CSDL pubs tôi cần thêm ít nhất là quyền gì? tại sao?

# Chương 3: Quản trị CSDL trong SQL Server

- Bài 4:** Dùng câu lệnh trong SQL Query Analyzer tạo một CSDL tên là Tink5 gồm hai file Tink5\_Data.mdf và Tink5\_Log.ldf chứa trong thư mục C:\database\fanmem. Kích thước ban đầu của hai file lần lượt là 2MB và 1MB, tốc độ tăng là 1MB, kích thước tối đa của file Tink5\_Data.mdf và Tink5\_Log.ldf lần lượt là 20MB và 10MB.
- Sau đó dùng câu lệnh trong SQL Query Analyzer đổi tên CSDL trên thành Tin5b\_5c.
  - Cuối cùng dùng câu lệnh trong SQL Query Analyzer xóa CSDL trên.



# Chương 3: Quản trị CSDL trong SQL Server

**Bài 5:** Có ba đăng nhập trong Windows là 5b, 5c, 5d, Tên máy chủ là HONGDUC. Dùng câu lệnh trong SQL Query Analyzer để tạo ba đăng nhập vào SQL Server dùng chính các tài khoản trong Windows này luôn (chế độ một tài khoản). Dùng tiếp câu lệnh gỡ bỏ đăng nhập 5d.

- Giả sử đã có CSDL Tink5, viết câu lệnh trong SQL Query Analyzer cho phép hai đăng nhập 5b, 5c ở trên dùng được CSDL này.
- Dùng tiếp câu lệnh gỡ bỏ đăng nhập 5b không cho phép dùng CSDL Tink5 nữa.

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 6:** Có một nhóm người dùng tên là M trong Windows gồm ba thành viên A, B, C. Tên máy chủ là HONGDUC.

- Dùng câu lệnh cấp phép cho nhóm M được quyền đăng nhập vào SQL Server.
- Dùng tiếp câu lệnh gỡ bỏ quyền đăng nhập của A vào SQL Server nhưng vẫn giữ nguyên quyền đăng nhập vào SQL Server của nhóm M.

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 7:** Dùng câu lệnh trong SQL Query Analyzer để tạo hai đăng nhập vào SQL Server ở chế độ hai tài khoản (SQL Server – authentication) với cặp username và password lần lượt là (5b, bb) và (5c, cc)

Sau đó đổi password của 5c thành 5ccc. Tiếp theo cấp phép cho 5c được dùng CSDL Tink5.

Dùng tiếp câu lệnh gỡ bỏ đăng nhập của 5b.

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 8:** Bạn hiểu vai trò (role) là gì? Có mấy mức, là những mức nào? Tại sao lại nên có vai trò?

**Bài 9:** Giả sử ta có hai tài khoản đăng nhập trong Windows là test1 và test2, tên máy chủ là HONGDUC. Dùng câu lệnh hãy cấp phép cho hai tài khoản này đăng nhập vào SQL Server ở chế độ một tài khoản, sau đó gán cho đăng nhập test1 là thành viên của vai trò dbcreator, test2 là thành viên của vai trò SecurityAdmin, và dbcreator. Sau đó gỡ bỏ test2 khỏi thành viên của vai trò dbcreator.

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 10:** Thế nào là vai trò ở mức CSDL?

- Có mấy vai trò cố định ở mức CSDL?
- Có thể xóa sửa được các vai trò này không?
- Kể tên và tác dụng của một số vai trò trong số đó?

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 11:** Trong CSDL Tink5, dùng câu lệnh hãy tạo vai trò tên là K5, sau đó dùng tiếp câu lệnh gán cho vai trò này hai quyền tạo bảng và tạo view.

Giả sử có hai người dùng trong CSDL Tink5 là 5b và 5c.

- a. Dùng câu lệnh để gán hai người dùng này là thành viên của vai trò K5.
- b. Dùng câu lệnh vô hiệu quyền tạo bảng của 5b, nhưng giữ nguyên quyền tạo bảng và tạo view của vai trò K5, và 5b vẫn là thành viên của vai trò K5.
- c. Xóa vai trò K5.

# Chương 3: Quản trị CSDL trong SQL Server

- Bài 12:** Giả sử có hai người dùng trong CSDL Tink5 là test1 và test2. Dùng câu lệnh hãy cấp quyền tạo bảng và tạo view cho hai tài khoản người dùng này. Dùng tiếp câu lệnh gỡ bỏ quyền tạo view của người dùng test1.
- Bài 13:** Bạn hiểu thế nào là quyền phát biểu (Statement Permissions) và quyền về đối tượng (Object Permissions)?

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 14:** Bạn hiểu thế nào là người sở hữu CSDL dbo?

- Thế nào là người sở hữu đối tượng dbo?
- Tại sao trong một CSDL lại nên để các đối tượng được sở hữu duy nhất bởi người dbo?



# Chương 3: Quản trị CSDL trong SQL Server

**Bài 15:** Giả sử có hai người dùng trong CSDL Tink5 là test1 và test2.

- Hãy dùng câu lệnh cấp cho người dùng test1 quyền select và insert trên bảng Canbo.
- Giả sử bảng Canbo có các trường là (Hoten, Tuoi, Gioitinh, Quequan, Chucvu, Luong), dùng câu lệnh cấp cho test2 quyền select trên các cột (Hoten, Tuoi, Gioitinh, Quequan, Chucvu), và quyền Update trên các cột (Hoten, Tuoi, Gioitinh, Quequan) và cho phép test2 quyền được cấp phép cho các người dùng khác quyền Update này.
- Gỡ bỏ quyền Update trên bảng Canbo của test2 và tất cả những người đã được test2 cấp cho quyền này.
- Vô hiệu hóa quyền select và insert trên bảng Canbo của test1.

# Chương 3: Quản trị CSDL trong SQL Server

## 3.3.2 Lời giải

### Bài 4:

- Tạo CSDL Tink5

*Use master*

*go*

*Create Database Tink5 on Primary (Name = Tink5\_Data,*

*Filename = 'C:\database\fanmem\Tink5\_Data.mdf',*

*Size = 2MB,*

*Maxsize = 20MB,*

*Filegrowth = 1MB)*

# Chương 3: Quản trị CSDL trong SQL Server

*Log ON*

*(Name = Tink5\_Log,*

*FileName='C:\database\fanmem\Tink5\_Data.ldf',*

*size = 1MB,*

*maxsize = 10MB,*

*Filegrowth = 1MB)*

*Go*

# Chương 3: Quản trị CSDL trong SQL Server

- Đổi tên CSDL Tink5 thành Tin5b\_5c:

```
use Tink5
```

```
Go
```

```
exec sp_dboption Tink5, 'Single User',true
```

```
exec sp_renamedb 'Tink5', 'Tin5b_5c'
```

```
exec sp_dboption Tin5b_5c, 'Single User',false
```

```
Go
```

```
Xóa CSDL Tin5b_5c:
```

```
Use master
```

```
Go
```

```
Drop Database Tin5b_5c
```

```
Go
```

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 5:** Cấp phép cho ba đăng nhập 5b, 5c, 5d  
(chế độ một tài khoản)

*Exec sp\_grantlogin 'HONGDUC\5b'*

*Exec sp\_grantlogin 'HONGDUC\5c'*

*Exec sp\_grantlogin 'HONGDUC\5d'*

Gỡ bỏ đăng nhập 5d:

*Exec sp\_revokellogin 'HONGDUC\5d'*

# Chương 3: Quản trị CSDL trong SQL Server

Cấp phép dùng CSDL Tink5 cho hai đăng nhập 5b và 5c:

*Use Tink5*

*Exec sp\_grantdbaccess 'HONGDUC\5b'*

*Exec sp\_grantdbaccess 'HONGDUC\5c'*

Gỡ bỏ người dùng 5b khỏi CSDL Tink5:

*Use Tink5*

*Exec sp\_revokedbaccess 'HONGDUC\5b'*

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 6:** Cấp phép cho nhóm M được đăng nhập vào SQL Server

*Exec sp\_grantlogin 'HONGDUC\M'*

Gỡ bỏ quyền đăng nhập của A

*Exec sp\_denylogin 'HONGDUC\A'*

**Bài 7:** Tạo hai đăng nhập:

*Exec sp\_addlogin '5b','bb'*

*Exec sp\_addlogin '5c','cc'*

# Chương 3: Quản trị CSDL trong SQL Server

Đổi password của 5c

*Exec sp\_password 'cc', '5ccc', 'username'*

Cấp phép cho 5c được dùng CSDL Tink5

*User Tink5*

*Exec sp\_grantdbaccess '5c'*

Gỡ bỏ đăng nhập 5b

*Exec sp\_droplogin '5b'*



# Chương 3: Quản trị CSDL trong SQL Server

**Bài 9:** Cấp phép cho test1 và test2 đăng nhập vào SQL Server ở chế độ một tài khoản:

```
Exec sp_grantlogin 'HONGDUC\test1'
```

```
Exec sp_grantlogin 'HONGDUC\test2'
```

Gán vai trò cho test1 và test2

```
Exec sp_addsrvrolemember 'HONGDUC\test1',  
    'dbcreator'
```

```
Exec sp_addsrvrolemember 'HONGDUC\test2',  
    'dbcreator'
```

```
Exec sp_addsrvrolemember 'HONGDUC\test2',  
    'SecurityAdmin'
```

# Chương 3: Quản trị CSDL trong SQL Server

Gỡ bỏ test2 khỏi vai trò dbcreator

```
Exec sp_dropsrvrolemember 'HONGDUC\test2', 'dbcreator'
```

**Bài 11:** Tạo vai trò K5 trong CSDL Tink5:

```
Use Tink5
```

```
Exec sp_addrole 'K5'
```

Gán quyền cho vai trò K5

```
Use Tink5
```

```
Grant Create table, create view to K5
```

# Chương 3: Quản trị CSDL trong SQL Server

Gán 5b và 5c là thành viên của K5

*Use Tink5*

*Exec sp\_addrolemember 'K5', '5b'*

*Exec sp\_addrolemember 'K5', '5c'*

Vô hiệu quyền tạo bảng của 5b

*Use Tink5*

*deny Create table to 5b.*

Xóa vai trò K5

*Use Tink5*

*Exec sp\_droprole 'K5'*

# Chương 3: Quản trị CSDL trong SQL Server

## Bài 12:

Cấp quyền cho hai người dùng test1 và test2

*Use Tink5*

*Grant Create table, create view to test1, test2*

Gỡ bỏ quyền tạo view của người dùng test1

*Use Tink5*

*Revoke Create view to test1*

# Chương 3: Quản trị CSDL trong SQL Server

**Bài 15:** Cấp cho người dùng test1 quyền select và insert trên bảng Canbo:

*Use Tink5*

*Grant select, insert on Chucdanh to test1.*

Cấp quyền select (Hoten, Tuoi, Gioitinh, Quequan, Chucvu), Update (Hoten, Tuoi, Gioitinh, Quequan) cho test2 và cho phép test2 quyền được cấp phép cho các người dùng khác quyền Update này.

*Grant Select on Canbo (Hoten, Tuoi, Gioitinh, Quequan, Chucvu) to test2*

*Grant Update on Canbo (Hoten, Tuoi, Gioitinh, Quequan) to test2 With Grant Option*

# Chương 3: Quản trị CSDL trong SQL Server

Gỡ bỏ quyền Update trên bảng Canbo của test2 và tất cả những người đã được test2 cấp cho quyền này

*Revoke Update on Canbo (Hoten, Tuoi, Gioitinh, Quequan) from test2 Cascade.*

Vô hiệu hóa quyền select và insert trên bảng Canbo của test1

*Deny select, insert on Canbo to test1*

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.1 Khái niệm về bảng

- SQL Server làm việc trên mô hình CSDL quan hệ, nên bảng là nơi chứa dữ liệu thực sự của CSDL.
- Bảng có các thành phần chính là trường (cột-column), bản ghi (dòng dữ liệu), khóa chính có thể có hoặc không, mỗi trường thì có một kiểu dữ liệu nào đó tùy thuộc từng HQTCSDL.
- Một CSDL có thể chứa đến 2 tỉ bảng, mỗi bảng có thể chứa tới 1024 cột, các cột phải được định danh duy nhất trong một bảng theo một quy tắc đặt tên (không dài quá 128 kí tự, ....).

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.2 Các kiểu dữ liệu

- **Loại String:** Bao gồm các kiểu dữ liệu như char(n), varchar(n), text, nchar, nvarchar, ntext. Trong đó nchar, nvarchar, ntext lưu trữ dữ liệu theo dạng Unicode (mỗi dữ liệu được lưu trữ trong 2 byte).

Nếu có thông số đằng sau thì chỉ rõ độ dài lớn nhất của chuỗi đó, thiếu thì thêm ký tự trắng vào, thừa thì cắt đi. Nếu không chỉ rõ thì chiều dài của chuỗi là tự tăng khi các ký tự trong chuỗi tăng lên, nhưng điều này trả giá là tốc độ xử lý chậm.



# Chương 4: Bảng (Table) và Lược đồ (Diagram)

Ngoài ra nếu chiều dài của chuỗi mà vượt quá 8000 ký tự thì phải dùng kiểu text.

Nếu dùng kiểu text thông thường trong thực tế khi lập trình người ta thường lưu trữ thành một file đặt ở một nơi nào đó trên ổ cứng vật lý còn trong CSDL người ta lưu trữ đường dẫn đến file đó.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- **Kiểu dữ liệu số chính xác:** Bao gồm các kiểu dữ liệu như bit, tinyint, smallint, int, bigint trong đó bit có chiều dài là 1 byte nhận hai giá trị là 0 hoặc 1, tinyint có chiều dài là 1 byte (0 – 255), smallint có chiều dài là 2 bytes (+ – 32767), int có độ dài là 4 bytes (+ – 214783657), bigint có chiều dài là 8 bytes (+ –  $2^{63}$ ).

Ngoài ra còn hai loại nữa là decimal và numeric, khi khai báo thì phải khai báo độ chính xác của phía phải dấu chấm thập phân (scale), mặc định ban đầu là 0.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- **Kiểu dữ liệu số gần đúng:** Bao gồm các kiểu dữ liệu real, float, trong đó real yêu cầu 4 bytes nên lưu trữ 7 ký số, còn float 8 bytes nên có thể lưu trữ lên tới 38 ký số.
- **Kiểu dữ liệu Date và Time:** Bao gồm hai loại Datetime và Smalldatetime. Trong đó Datetime gồm 8 bytes lưu trữ phạm vi từ 1/1/1753 – 12/31/9999, Smalldatetime gồm 4 bytes lưu trữ phạm vi từ 1/1/1900 đến 6/6/2079.

Trong Datetime giờ được lưu trữ đến miliseconds, trong smalldatetime được lưu trữ đến từng phút.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

Lưu ý: Nếu chỉ cần làm việc đến năm, tháng, ngày nên dùng kiểu dữ liệu là số thì sẽ dễ dàng hơn trong quá trình lập trình. Ví dụ để lưu trữ năm ta dùng kiểu dữ liệu smallint, ngày trong tháng dùng tinyint...

- **Kiểu tiền tệ:** Bao gồm hai loại là money và smallmoney, cả hai đều lưu trữ bốn chữ số ở bên phải dấu thập phân, khi nhập dữ liệu tiền tệ phải nhập ở trước dấu \$. Trong đó money gồm 8 byte lưu trữ phạm vi từ + - 922337203685447.5808, smallmoney gồm 4 byte lưu trữ phạm vi từ + - 214748.3647.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- **Kiểu dữ liệu người dùng:** Người dùng có thể tự tạo kiểu dữ liệu do mình muốn dựa vào những dữ liệu do hệ thống cung cấp. Sử dụng thủ tục lưu trữ hệ thống `sp_addtype`, ví dụ:

*Use Pubs*

*Exec sp\_addtype test, 'char(9)', 'null'*

Câu lệnh này thêm kiểu dữ liệu `test` vào CSDL `Pubs`, kiểu dữ liệu này có chính là kiểu `char(9)`, và cho phép nhận giá trị `null` nếu không có giá trị mặc định.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

Có thể tạo bảng giao diện bằng cách bấm chuột phải vào User defined Data Types trong một CSDL nào đó mà mình muốn tạo kiểu dữ liệu mới sau đó chọn New User defined Data Types , các bước sau làm theo hướng dẫn của giao diện. Để xóa một kiểu dữ liệu người dùng trong giao diện chỉ đơn giản là vào mục User defined Data Types chọn kiểu dữ liệu cần xóa bấm chuột phải chọn delete, bằng câu lệnh thì chạy lệnh:

*Use Tên CSDL*

*Exec sp\_droptype tên kiểu dữ liệu cần xóa.*

Lưu ý rằng để xóa được kiểu dữ liệu người dùng nào đó thì yêu cầu kiểu DL đó phải không được dùng ở bất kỳ đâu.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.3 Làm việc với bảng

### 4.3.1 Tạo và sửa bảng

Hai cách dùng câu lệnh hoặc giao diện

- Dùng giao diện: Bấm chuột phải vào Table chọn new Table, trong cửa sổ giao diện, sau đó tạo mới các trường và chọn kiểu dữ liệu tương ứng cần thiết. Trong nửa dưới màn hình bạn có thể thiết lập các thuộc tính, chẳng hạn như các diễn giải, các giá trị mặc định, thuộc tính identity, ... Khi hoàn tất click vào nút Save và đặt tên cho bảng.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Để sửa bảng chỉ đơn giản là bấm chuột phải vào tên bảng cần sửa và chọn Design Table, thực hiện công việc sửa và chọn Save.
- Dùng câu lệnh: *Create Table Tên bảng*  
(*Tên trường kiểu dữ liệu Các thuộc tính,*  
*...*  
*)*

Ví dụ: *Create Table Test*  
( *id tinyint Identity Not Null,*  
*Hoten nvarchar(30) Null*)



# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.3.2 Xóa bảng:

Hai cách dùng câu lệnh hoặc giao diện

- Dùng giao diện: Bấm chuột phải vào Table chọn delete.
- Dùng câu lệnh: *Drop Table tên bảng*
- Chú ý: Các bảng hệ thống không thể bị xóa, các bảng mà còn đang bị ràng buộc cũng không thể bị xóa.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.4 Các loại toàn vẹn

Có ba loại toàn vẹn:

- Toàn vẹn trên miền: Các loại dữ liệu giúp xác định các giá trị gì là hợp lệ cho một cột riêng biệt nào đó điều này được gọi là toàn vẹn miền (domain integrity). Miền đơn giản là một tập các giá trị hợp lệ cho một cột riêng biệt.
- Toàn vẹn tham chiếu: Toàn vẹn tham chiếu liên quan đến việc quản lý các mối quan hệ giữa các dòng dữ liệu trong nhiều bảng.
- Toàn vẹn thực thể: Có nghĩa là bạn có thể nhận diện duy nhất mỗi dòng dữ liệu trong một bảng, đây chính là ràng buộc khóa chính.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.5 Các phương pháp toàn vẹn

### 4.5.1 Các phương pháp truyền thống

#### 4.5.1.1 Dùng các kiểu dữ liệu do người dùng định nghĩa

- Chọn một kiểu dữ liệu cụ thể cho một trường của bảng, dữ liệu được nhập vào trường này sẽ nằm trong miền giới hạn của kiểu dữ liệu đó.
- Ví dụ tạo kiểu dữ liệu mới *test*

*Use Pubs*

*Exec sp\_addtype test, 'char(9)', 'null'*

- Trong CSDL Pubs tạo mới một bảng và chọn kiểu dữ liệu của trường trong bảng đó là *test*, lúc này dữ liệu được nhập vào trường này sẽ có giới hạn là *char(9)*.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.5.1.2 Các mặc định (Default Value)

- Giá trị mặc định mà một cột của trường sẽ nhận được khi dữ liệu của cột ấy không được thêm vào ở câu lệnh Insert.
- Để thực hiện điều này mở bảng ở chế độ Design Table, chọn trường cần thiết lập mặc định, sau đó ở nửa dưới màn hình chọn mục Default Value và nhập giá trị mặc định.
- Lưu ý giá trị mặc định phải nằm trong miền giá trị cho phép của kiểu dữ liệu của trường mà bạn muốn thiết lập mặc định.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.5.1.3 Các quy tắc (Rules)

- Các quy tắc cũng cố thêm tính toàn vẹn miền bằng cách cung cấp nhiều hơn việc kiểm tra các giá trị hợp lệ. Các quy tắc được sử dụng đảm bảo các giá trị, dùng các biểu thức của ngôn ngữ SQL.
- Ví dụ để tạo mới một quy tắc (Rule), mở CSDL mà ta muốn tạo Rule, bấm chuột phải vào Rules chọn New Rule, trong hộp Name gõ tên Rule mà bạn muốn tạo, ở hộp text bạn gõ cấu trúc: *@tên biểu thức SQL*. Sau đó bấm OK.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Ví dụ muốn tạo một Rule tên là myRule mà có quy tắc là giá trị bé hơn 100:  
ở hộp Name ta gõ: myRule  
ở hộp text ta gõ: @tt < 100
- Nếu ta muốn tạo một Rule có tên là Tink5 mà có quy tắc là những ký tự đầu tiên phải bắt đầu bằng Tink5:  
ở hộp Name ta gõ: Tink5  
ở hộp text ta gõ: @kk Like 'Tink5%'

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Để gán một cột nào đó nhận một quy tắc nào đó ta làm như sau: Trong cửa sổ Rules bên phải, bấm chuột phải vào quy tắc cần gán, chọn Properties, ở cửa sổ Rule Properties chọn Bind Columns, ở mục Table, chọn bảng mà chứa cột cần gán, sau đó chọn cột cần gán ở phía trái và bấm Add, tương tự muốn bỏ gán tới một cột nào đó ta chọn cột đó và bấm Remove.
- Lưu ý là kiểu dữ liệu của cột cần gán và biểu thức của quy tắc (Rule) là phải phù hợp với nhau.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Ví dụ bảng Vidu2 có cột tên là gg có kiểu tinyint, ta muốn yêu cầu dữ liệu nhập vào cột này phải nhỏ hơn 100, tức là ta gán cho cột này nhận quy tắc myRule ta làm như sau:
- Bấm chuột phải vào myRule, chọn Properties, chọn Bind Columns, ở mục Table chọn bảng Vidu2, chọn cột gg bên trái và bấm Add, cuối cùng là bấm OK.



# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.5.2 Lược đồ và sự toàn vẹn

Ta sẽ thực hiện các loại toàn vẹn bằng giao diện:

**4.5.2.1 Thực hiện toàn vẹn thực thể:** Mở bảng ở chế độ Design Table, để thiết lập khóa chính cho một bảng chọn cột hoặc các cột mà bạn muốn thiết lập khóa chính rồi click vào biểu tượng chìa khóa màu vàng trên thanh công cụ để thiết lập khóa chính. Yêu cầu khóa chính xác định duy nhất một dòng dữ liệu trong bảng.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Nếu không có khóa chính tự nhiên ta có thể dùng thuộc tính Identity (khóa chính tự tăng).
- Nếu dùng Identity có hai thuộc tính là seed và Increment, seed là giá trị tăng bắt đầu, Increment là bước tăng. Ví dụ nếu seed = 2 và Increment = 1 có nghĩa là bắt đầu tăng từ 2 và mỗi lần tăng lên 1.
- Để chuyển giá trị của Identity về lại giá trị seed ban đầu ta dùng câu lệnh

*Truncate Table Tên bảng*

- Câu lệnh này sẽ xóa toàn bộ các dòng dữ liệu trong bảng và chuyển giá trị của trường khóa Identity về giá trị seed ban đầu.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

**4.5.2.2 Thực hiện toàn vẹn miền:** Chọn một kiểu dữ liệu cụ thể cho một trường của bảng, dữ liệu được nhập vào trường này sẽ nằm trong miền giới hạn của kiểu dữ liệu đó.

**4.5.2.3 Thực hiện toàn vẹn tham chiếu:** Ta dùng lược đồ (Diagram), mở CSDL mà bạn muốn làm việc, bấm chuột phải vào Diagrams chọn New Database Diagram, bấm tiếp Next, chọn những bảng cần thực hiện toàn vẹn tham chiếu và bấm Add, bấm tiếp Next và cuối cùng bấm Finish.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Ở màn hình làm việc của Diagrams chọn cột của bảng muốn làm quan hệ khóa chính giữ chuột và kéo tới bảng mà ta muốn có quan hệ (cột của bảng này sẽ xem như là khóa ngoại), lúc này sẽ xuất hiện màn hình Create Relationship, chọn các cột tương ứng ở hai bảng, bảng bên trái (ở mục Primary Key Table) sẽ chứa cột là khóa chính, còn bên phải chứa cột là khóa ngoại.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

- Nếu tích vào ô Cascade Update Related Fields có nghĩa là cập nhật ở bảng chính thì cũng cập nhật liên hoàn ở các bảng phụ (các bảng chứa khóa ngoại), còn nếu chọn Cascade Delete Related Records, thì có nghĩa là nếu xóa ở bảng chính thì cũng sẽ xóa tương ứng ở các bảng phụ.
- Lưu ý là các cột khóa chính, khóa ngoại phải có kiểu dữ liệu phù hợp với nhau.
- Để xóa ràng buộc chỉ đơn giản là bấm chuột phải vào ràng buộc và chọn Delete Relationship From Database.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

## 4.6 Bài tập

**Bài 1:** Có cần sử dụng các kiểu dữ liệu do người dung định nghĩa không? Một số quy tắc thông thường phải tuân theo khi thao tác với giá trị số? Một số quy tắc thông thường phải tuân theo khi thao tác với dữ liệu kiểu ký tự? Điểm quan trọng nhất phải nhớ khi tạo các bảng là gì?

**Bài 2:** Tạo một bảng chứa thông tin về cửa hàng, dữ liệu nên lưu trữ bao gồm định danh cửa hàng, tên, địa chỉ, thành phố, bang, ZIP, chủ nhân, người liên hệ, số FAX và Email.

# Chương 4: Bảng (Table) và Lược đồ (Diagram)

**Bài 3:** Tạo một bảng chứa thông tin về việc bán hàng, dữ liệu nên lưu trữ bao gồm định danh cửa hàng, ngày bán, tổng số đã bán, tổng số tiền phải trả và số tiền đặt cọc.

**Bài 4:** Bạn sẽ tạo các kiểu dữ liệu do người dùng định nghĩa sau như thế nào?

- Zip\_Code
- Phone\_Number
- Store\_id
- Fax
- Email

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.1 Làm việc với truy vấn

- Mục tiêu của truy vấn là truy xuất thông tin từ cơ sở dữ liệu, có một số môi trường làm việc như MSQuery, OSQL, SQL English Query nhưng ta quan tâm ở đây là môi trường SQL Query Analyzer.
- Câu lệnh cơ bản của một câu truy vấn là:

Select <column\_list>

From <table\_list>

Where <search\_criteria>



# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Bạn cũng có thể thay đổi tiêu đề cột cho bằng cách đặt bí danh cho nó, ví dụ như:

```
Select maChucdanh as Mã_chức_danh,  
TenChucdanhDD as Tên_Chức_danh_đầy_đủ  
From Chucdanh
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Bạn cũng có thể dùng Literals để cho ra kết quả dễ đọc hơn, Literal là một chuỗi ký tự được bao quanh những ( ' ' ) trong column\_list và được hiển thị như một cột khác trong kết quả truy vấn. Kết quả xuất hiện có một mác cạnh cột thông tin. Ví dụ:

```
Select maChucdanh, 'Loai chuc danh tong quat:',  
maloaichucdanh
```

```
From Chucdanh
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.2 Các hàm và toán tử trong truy vấn

### 5.2.1 Các hàm toán học

- Các hàm toán học cho phép bạn thực hiện các phép toán cần thiết thông thường trên dữ liệu toán học. Bạn có thể trả về dữ liệu toán học bằng cách sử dụng cú pháp sau:
- Cú pháp: `Select Function_name (Parameters)`
- Bảng sau liệt kê một số các hàm cơ bản:

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

Hàm	Kết Quả
ABS(numric_expr)	Giá trị tuyệt đối
ACOS   ASIN   ATAN (Float_expr)	Góc theo radian đối số của hàm cosine, sine hoặc tanget là một giá trị theo kiểu dữ liệu float
COS   SIN   COT   TAN (float_expr)	Là số cosine, sin, hoặc tangent của góc (theo radian)
CEILING (numeric_expr)	Số nguyên nhỏ nhất lớn hơn hoặc bằng với giá trị xác định
DEGREES (numeric_expr)	Chuyển đổi từ radian sang độ

EXP (float_expr)	Giá trị mũ của giá trị xác định
FLOOR (numeric_expr)	Số nguyên lớn nhất nhỏ hơn hoặc bằng giá trị xác định
LOG(float_expr)	Số log tự nhiên
LOG10 (float_expr)	Log cơ số 10
PI ( )	Hằng số 3.141592653589793
POWER (numeric_expr, y)	Giá trị của numeric_expr lũy thừa y
RADIAN (numeric_expr)	Chuyển đổi từ độ sang radian
RAND([seed])	Số float ngẫu nhiên từ 0 đến 1
ROUND (numeric_expr, len)	Số numeric_expr làm tròn đến chiều dài xác định trong một số nguyên
SIGN (numeric_expr)	Số dương, âm hoặc bằng 0
SQUARE (float_expr)	Bình phương giá trị xác định
SQRT (float_expr)	Căn của giá trị xác định

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

Ví  
dụ  
:

Phát biểu	Kết quả
SELECT SQRT (9)	3.0
SELECT ROUND (1234.56, 0)	1235
SELECT ROUND (123.56, 1)	123.60
SELECT ROUND (\$1234.56, 1)	1,234.60
SELECT POWER (2, 8)	256.0
SELECT ABS (- 365)	365
SELECT SQUARE (3)	9

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Khi bạn dùng kiểu dữ liệu float có thể bạn nhận những kết quả không mong đợi. Nên nói chung bạn nên tránh dùng kiểu dữ liệu float.

## 5.2.2 Các hàm chuỗi

- Khi bạn thao tác với thông tin ký tự, bạn có thể sử dụng các hàm chuỗi khác nhau để thao tác dữ liệu.
- Cú pháp: `Select function_name (parameters)`

Hàm	Kết quả
'Expression' + 'Expression'	Kết nối hai hay nhiều chuỗi ký tự.
ASCII (char_expr)	Trả về giá trị mã ASCII của ký tự sát bên trái.
CHAR (integer_expr)	Trả về ký tự tương ứng với giá trị mã ASCII
DIFFERENCE (char_expr1, char_expr2)	So sánh hai chuỗi và đánh giá sự giống nhau; trả về một giá trị từ 0 đến 4, 4 là giống nhau nhất
LEFT (char_expr, integer_expr)	Trả về một chuỗi ký tự bắt đầu từ bên trái và đứng trước integer_expr ký tự
LOWER (char_expr)	Chuyển thành chữ thường
LTRIM (char_expr)	Trả về dữ liệu không có khoảng trắng đường trước
NCHAR (integer_expr)	Trả về ký tự theo mã Unicode tương ứng với integer_expr



<b>PATINDEX ('%pattern%',Expression)</b>	<b>Trả về vị trí bắt đầu của lần xuất hiện đầu tiên pattern trong expression</b>
<b>REPLACE ('string1', 'string2', 'string3')</b>	<b>Thay thế tất cả lần xuất hiện string2 trong string1 bằng string3</b>
<b>REPLICATE (char_expr, integer_expr)</b>	<b>Lặp lại chuỗi char_expr với integer_expr lần</b>
<b>REVERSE (char_expr)</b>	<b>Trả về chuỗi đảo ngược của char_expr</b>
<b>RIGHT (char_expr, integer_expr)</b>	<b>Trả về một chuỗi ký tự bắt đầu từ integer_expr ký tự tính từ bên phải</b>
<b>RTRIM (char_expr)</b>	<b>Trả về dữ liệu không có khoảng trắng ở đằng sau</b>
<b>SOUNDEX (char_expr)</b>	<b>Trả về mã bốn chữ số để tính mức tương đương của hai chuỗi ký tự</b>
<b>SPACE (integer_expr)</b>	<b>Trả về một chuỗi lặp lại integer_expr khoảng trắng</b>
<b>STUFF (char_expr1,start,length, char_expr2)</b>	<b>Xoá length ký tự khỏi char_expr1 ở vị trí đầu và chèn char_expr2 vào đó</b>
<b>SUBSTRING (expression, start, length)</b>	<b>Trả về một phần chuỗi ký tự hoặc chuỗi nhị phân</b>
<b>UNICODE ('nchar_string')</b>	<b>Trả về giá trị số theo mã Unicode của ký tự đầu tiên trong 'nchar_expr'</b>
<b>UPPER (char_expr)</b>	<b>Chuyển thành chữ hoa</b>

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

Ví dụ:

Phát biểu	Kết quả
SELECT ASCII ('G')	71
SELECT PATINDEX ('%BC%', 'ABCDE')	2
SELECT RIGHT ('ABCDE', 3)	CDE
SELECT REVERSE ('ABCDE')	EDCBA
SELECT LOWER ('ABCDE')	abcde

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.2.3 Các hàm ngày tháng

- Bạn có thể biến đổi các giá trị thuộc kiểu dữ liệu datetime bằng cách sử dụng các hàm ngày tháng.
- **Cú pháp:** `Select date_function (parameters)`
- Bạn phải đặt các giá trị thuộc kiểu dữ liệu datetime truyền vào cho các tham số giữa các dấu nháy đơn hoặc dấu nháy đôi. Một số hàm sử dụng một thông số gọi là `datepart`. Bảng sau liệt kê các giá trị của `datepart` và chữ viết tắt của chúng.

<b>Datepart</b>	<b>Viết tắt</b>	<b>Giá trị</b>
Day	dd	1 – 31
Day of Year	dy	1 – 366
Hour	Hh	0 – 23
Millisecond	Ms	0 – 999
Minute	Mi	0 – 59
Month	Mm	1 – 12
Quarter	Qq	1 – 4
Second	Ss	0 – 59
Week	Wk	0 – 53
Weekday	Dw	1 – 7 (Sun – Sat)
Year	Yy	1753 – 9999

## Các hàm ngày tháng:

Hàm	Kết quả
DATEDD (datepart, number, date)	Thêm số datepart vào date
DATEDIFF (datepart, date1, date2)	Trả về số datepart giữa hai ngày
DATENAME (datepart, date)	Trả về giá trị ASCII cho giá trị của datepart của ngày date
DATEPART (datepart, date)	Trả về giá trị integer cho giá trị của datepart của ngày date
DAY (date)	Trả về một giá trị số nguyên thể hiện ngày
GETDATE ( )	Trả về ngày và giờ hiện hành
MONTH (date)	Trả về một giá trị số nguyên thể hiện tháng
YEAR (date)	Trả về một giá trị số nguyên thể hiện năm

## Ví dụ:

Hàm	Kết quả
SELECT DATEDIFF (mm, '1/1/00', '12/31/02')	35 – (tháng)
SELECT GETDATE ( )	Apr 27, 2004 2:32 AM
SELECT DATEDDD (mm, 6, '1/1/04')	Jul 1. 2004 2:32 AM
SELECT DATEDDD (mm, -5, '10/6/04')	May 6, 2004 2:32 AM

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.2.4 Các hàm tập hợp

Hàm	Kết quả
AVG ([ALL   DISTINCT ] column_name)	Trả về trị trung bình của các giá trị trong biểu thức số, toàn bộ hoặc riêng biệt
COUNT (*)	Trả về số dòng được chọn
COUNT ([ALL   DISTINCT ] column_name)	Trả về số các giá trị trong biểu thức, tất cả hoặc riêng biệt
MAX (column_name)	Trả về giá trị lớn nhất trong biểu thức

MIN (column_name)	Trả về giá trị nhỏ nhất trong biểu thức
STDEV (column_name)	Trả về độ lệch chuẩn thống kê của tất cả các giá trị trong tên cột hoặc biểu thức
STDEVP (column_name)	Trả về độ lệch chuẩn thống kê đối với mẫu của tất cả các giá trị trong tên cột hoặc biểu thức
SUM ([ALL   DISTINCT] column_name)	Trả về tổng số các giá trị trong biểu thức số, tất cả hoặc riêng biệt
TOP n [PERCENT ]	Trả về n giá trị đầu hoặc n% giá trị trong tập kết quả của bạn
VAR (column_name)	Trả về sự dao động của các giá trị liệt kê trong tên cột hoặc biểu thức <sup>112</sup>



# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3 Xây dựng điều kiện trong truy vấn

- Cú pháp: `Select column_list`  
`From table_list`  
`Where search_conditions`

### 5.3.1 Các phép toán so sánh

- Bạn có thể thực hiện điều kiện bằng các phép toán so sánh như '=', '>', '<', '!=', ...
- Ví dụ: `Select * from chucdanh where machucdanh = 1`

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.2 Phạm vi

- Bạn có thể trích xuất các dòng dựa trên phạm vi của các giá trị bằng cách sử dụng từ khoá BETWEEN.
- Cú pháp: 

```
Select column_list  
      From table_list  
      Where column_name [NOT] BETWEEN  
      expression AND expression
```
- Ví dụ: 

```
select * from sinhvien where ngaysinh  
between '1/1/1976' and '2/2/2000'
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.3 Danh sách

- Bạn có thể trích xuất các dòng với các giá trị mà so khớp chúng trong một danh sách bằng cách sử dụng từ khoá IN.
- Cú pháp: 

```
Select column_list  
From table_list  
Where column_name [NOT] IN (value_list)
```
- Ví dụ: 

```
select * from hocnganh2  
where maNganh in (  
select maNganh from nganh where  
maDonvi=1)
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.4 Chuỗi ký tự

- Bạn có thể trích xuất các dòng dựa trên các phần của các chuỗi ký tự bằng cách sử dụng từ khoá LIKE.
- Cú pháp: 

```
Select column_list  
From table_list  
Where column_name [NOT] LIKE 'string'
```
- Các ký tự thay thế là:
  - % : Chuỗi không có ký tự hay nhiều ký tự
  - : Một ký tự.
  - [] : Một ký tự bên trong phạm vi xác định.
  - [^] : Một ký tự không nằm bên trong phạm vi xác định.
- Ví dụ: 

```
Select * from sinhvien where hosinhvien Like 'Lê%'  
Select * from sinhvien where hosinhvien Like '[VT]%'
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.5 Các giá trị không xác định

- Một giá trị NULL không giống với các giá trị 0 cũng như khoảng trắng, giá trị NULL có nghĩa là unknown. Trong các trường hợp như vậy một giá trị NULL sẽ không thể tham gia vào các phép so sánh ở trên. Lúc này ta làm việc với giá trị NULL bằng cách kiểm tra IS NULL hoặc IS NOT NULL.
- Ví dụ: `Select * from sinhvien Where anhcanhan IS NULL`

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.6 Giới hạn thông tin trùng

- Bạn có thể bỏ thông tin trùng lặp bằng cách dùng DISTINCT
- Ví dụ: `Select DISTINCT(gioitinh) from sinhvien`

## 5.3.7 Sắp xếp dữ liệu bằng cách sử dụng mệnh đề ORDER BY

- Bạn có thể sắp xếp kết quả truy vấn của bạn bằng cách sử dụng mệnh đề ORDER BY trong phát biểu Select của bạn

- Cú pháp: `Select column_list`

`From table_list`

`[ORDER BY column_name | column_list [ASC]`

`DESC]`

- Ví dụ: `Select * from sinhvien ORDER BY tensinhvien ASC`<sup>118</sup>

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.3.8 Compute

- Mệnh đề Compute sinh ra dòng dữ liệu tổng hợp mới, chúng sử dụng các hàm tập hợp được chỉ rõ trước đây.
- Cú pháp: *Select column\_list*  
*From table\_list*  
*Where search\_criterial*  
*[Compute] aggregate\_expression(s)*
- Ví dụ: *Select \* from canbo*  
*compute Sum(mahocvi)*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.4 Các loại truy vấn và Làm việc với các Truy vấn con

- Một phát biểu Select lồng bên trong phát biểu Select, Insert, Update hoặc Delete khác thường được xem là một truy vấn con (subquery).
- Nếu truy vấn con trả về một giá trị, chẳng hạn như một tổng số, nó có thể sử dụng ở bất kỳ nơi nào mà một giá trị đơn có thể được sử dụng.
- Nếu truy vấn con trả về một danh sách, chẳng hạn như một cột đơn của nhiều giá trị, nó chỉ có thể được sử dụng trong mệnh đề Where.



# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Truy vấn con luôn luôn được đặt bên trong các dấu ngoặc đơn, ngoại trừ bạn đang thực hiện một truy vấn liên kết, kết thúc trước khi truy vấn ngoài được thực hiện.
- Một truy vấn con có thể chứa các truy vấn con khác, không giới hạn số lượng các truy vấn con.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.4.1 Các loại truy vấn con

- Có hai loại, loại thứ nhất trả về một giá trị đơn, ta có thể dùng với các phép toán như: '=', '<', '>', '<>',....
- Loại thứ hai trả về một cột đơn hoặc nhiều giá trị ta có thể dùng phép toán so sánh của danh sách như IN và EXISTS trong mệnh đề Where.
- **Ví dụ:**

1. *select \* from hocnganh2*

*where maNganh in (*

*select maNganh from nganh where*

*maDonvi=1)*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

2. *select \* from hocnganh2*

*where maNganh < (select  
maLoaiChucdanh from chucdanh where  
maChucdanh = 29)*

3. *select \* from hocnganh2*

*where EXISTS*

*(select \* from Nganh where maNganh = 1)*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.4.2 Truy vấn con tương quan

- Một truy vấn con tương quan (correlated subquery) tham chiếu tới một bảng từ một truy vấn ngoài và ước lượng mỗi dòng đối với truy vấn ngoài.
- Trong khía cạnh này, một truy vấn tương quan biến đổi từ một truy vấn con bình thường bởi vì truy vấn con phụ thuộc vào các giá trị từ truy vấn bên ngoài.
- Một truy vấn bình thường được thực thi không lệ thuộc vào truy vấn bên ngoài.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- **Ví dụ:** Lấy tất cả những sinh viên sống cùng huyện với nhau và có cùng giới tính.

*Select \* from sinhvien sv1*

*Where gioitinh IN*

*(select gioitinh from sinhvien sv2*

*where sv1.maHuyen = sv2.maHuyen*

*And sv1.maSinhvien <> sv2.maSinhvien)*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.4.3 SELECT INTO

- Phát biểu Select Into cho phép bạn tạo một bảng mới dựa trên các kết quả truy vấn.
- Bảng mới bạn tạo dựa trên các cột bạn xác định trong danh sách chọn lựa, các bảng bạn nêu trong mệnh đề from và các dòng bạn chọn trong mệnh đề Where.
- Bạn có thể tạo hai loại bảng với phát biểu Select Into: thường trực và tạm thời.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- **Cú pháp:**

```
Select column_list  
INTO new_table_name  
From table_list  
Where search_criteria
```
- **Chú ý:** Có hai loại bảng tạm, thứ nhất là bảng tạm chỉ tồn tại và có giá trị trong phiên làm việc của người dùng hiện hành, nếu phiên làm việc của người dùng kết thúc bảng tạm sẽ bị xoá, ta tạo bảng tạm như thế bằng cách đặt # trước tên bảng.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Loại thứ hai là tồn tại toàn cục có giá trị trong tất cả các phiên làm việc hiện hành và sẽ bị xoá bỏ khi phiên làm việc của người dùng cuối cùng kết thúc, ta tạo bằng cách đặt ## trước tên bảng.
- Nếu tạo bảng thường trực bạn phải thiết lập tùy chọn SELECT INTO/ BULKCOPY của cơ sở dữ liệu, tên bảng định nghĩa phải là duy nhất trong CSDL và tuân theo quy tắc.



# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Ví dụ:

1. *Select hosinhvien, tensinhvien  
Into #sv1  
From sinhvien*

2. *Use DaoTaoHongDuc  
Go*

*Exec sp\_dboption 'DaoTaoHongDuc', 'Select  
into/bulkcopy', true*

*go*

*Select \* into tmptest from chucdanh*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.4.4 Toán tử Union

- Bạn có thể kết nối kết quả của hai hay nhiều truy vấn thành một tập kết quả đơn bằng cách sử dụng toán tử Union.
- Mặc nhiên hai dòng trùng bị giới hạn, tuy nhiên việc sử dụng Union với từ khoá All trả về tất cả các dòng bao gồm cả các dòng trùng.
- Tất cả các column\_list phải có cùng số cột, cùng trật tự cột và có cùng loại dữ liệu.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- **Cú pháp:**

Select column\_list From table\_list Where clause

...

Union [All]

Select column\_list From table\_list Where clause

...

- **Ví dụ:**

*Select tenhuyen, matinh from huyen*

*union*

*select null, matinh from tinh*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.5 Hiệu chỉnh dữ liệu

### 5.5.1 Chèn dữ liệu

- Phát biểu Insert cơ bản là một thời điểm bạn có thể thêm một dòng vào trong một bảng, bằng cách biến thể ta cũng có thể thêm nhiều dòng vào bảng bằng cách chọn dữ liệu từ các bảng khác.
- Ví dụ:

```
insert into uutiengiamtiet(maCanbo,  
maTruonghopUutienGT, namhoc, hocky)  
values(36, 1, 2006, 1)
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- **Chú ý:** Ta có thể không cần chỉ rõ tất cả các cột, nếu không chỉ rõ giá trị null (nếu cho phép null) hoặc giá trị mặc định (Default Value) sẽ được tự động thêm vào, nếu cả hai dạng này đều không có phát biểu sẽ sinh lỗi.
- Ta cũng có thể thêm cùng lúc nhiều dòng bằng cách dùng truy vấn con Select, nhưng phát biểu Select ở đây khác là không có dấu ngoặc đơn như trên
- **Ví dụ:**

```
Insert into tmpctest(tenchucdanhVT, tenchucdanhDD,  
maloaichucdanh)
```

```
select tenchucdanhVT, tenchucdanhDD, maloaichucdanh from  
chucdanh
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.5.2 Xoá dữ liệu

- Phát biểu delete đơn giản là xoá một hoặc nhiều dòng từ một bảng.

Ví dụ: *Delete from chucdanh where machucdanh = 1*

- Ngoài ra ta cũng có thể thực hiện phát biểu Delete bằng cách dùng bảng dò tìm (hay truy vấn con)

Ví dụ: *Delete from chucdanh*

*Where maloachucdanh in*

*(Select maloachucdanh from loaichucdanh)*

- Ngoài ra ta cũng có thể sử dụng lệnh Truncate Table.

Ví dụ: *Truncate Table chucdanh.*

Sẽ gỡ bỏ tất cả dữ liệu từ một bảng chỉ giữ lại cấu trúc của bảng.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.5.3 Cập nhật dữ liệu

- Sử dụng phát biểu Update đơn giản là cập nhật lại dữ liệu trên một hoặc nhiều cột ở một hoặc nhiều hàng trong một bảng.
- Ví dụ: *update Chucdanh set*

*tenChucdanhVT='HT',*

*tenChucdanhDD='Hieu Truong',*

*maLoaiChucdanh=1*

*where maChucdanh=30*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

- Ngoài ra cũng như trên ta cũng có thể sử dụng phát biểu Update bằng cách dùng bảng dò tìm hoặc truy vấn con.
- Ví dụ: *Update Chucdanh*  
*Set maloichucdanh =*  
*(select maloichucdanh from*  
*loaichucdanh where tenloaichucdanh =*  
*'dang')*



# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6 Sử dụng các phần mở rộng của T-SQL

### 5.6.1 Các bó (Batch)

- Đơn giản một bó là một tập các phát biểu T-SQL được biên dịch đồng thời bởi SQL Server, phát biểu GO đánh dấu kết thúc của bó.
- Ví dụ: 

```
Use Daotaohongduc
select * from chucdanh
select * from loaichucdanh
go
```
- Chú ý các phát biểu Create phải tạo thành các bó riêng không được liên kết với các phát biểu khác

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.2 Kịch bản (Script)

- Một kịch bản là một tập của một hoặc nhiều bó được lưu vào một nhóm
- Ví dụ: *Use Daotaohongduc*

*select \* from chucdanh*

*select \* from loaichucdanh*

*go*

*Select \* from canbo*

*Go*

- Cách tạo kịch bản đã được đề cập đến trong chương 3 mục 3.2.6.3

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.3 Chuyển tác (Transaction)

- Một chuyển tác là một đơn vị làm việc trong đó tất cả các phát biểu phải được xử lý thành công, hoặc là không có phát biểu nào được xử lý, đây là vấn đề rất quan trọng trong CSDL phân tán và Hệ phân tán,
- Chúng ta sẽ được nghiên cứu kỹ hơn về chuyển tác trong môn học CSDL nâng cao và Hệ phân tán.

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.4 Các thành phần của ngôn ngữ Control-of-flow

### 5.6.4.1 Khối Begin ... End

- Đơn giản như trong ngôn ngữ lập trình nếu bạn muốn thực hiện một tập các phát biểu được thực thi với nhau bạn đặt chúng giữa tập lệnh Begin ... End.

### 5.6.4.2 Phát biểu Print

- Từ trước đến giờ ta chỉ biết đến phát biểu Select dùng để trả về kết quả, Print cũng là một phát biểu dạng đó nhưng có sự giới hạn hơn.
- Ví dụ :           Print 'Hello'

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.4.3 Khối If ... Else

- Cũng như ngôn ngữ lập trình một khối If...Else là một phát biểu được thực thi tùy theo điều kiện.
- Ví dụ: *If (datename(dw, getdate())='Friday')*

*Begin*

*Print 'Thu 6'*

*Print 'Have a good weekend'*

*End*

*Else*

*Begin*

*Print 'Have a good day'*

*End*

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.4.4 Biểu thức Case

- Ý nghĩa cũng tương tự như trong ngôn ngữ lập trình.
- Ví dụ: 

```
Select macanbo , day = CASE  
    When khoiluong < 10 Then 'It'  
    When khoiluong Between 10 and 50 Then  
'Trung binh'  
    Else 'Nhieu Qua'  
End  
from thuchiencvkhacday
```

# Chương 5: Truy vấn và sử dụng các phần mở rộng của T-SQL

## 5.6.4.5 Sử dụng câu lệnh While

- Ý nghĩa cũng tương tự như trong ngôn ngữ lập trình.
- Ví dụ: *while (select avg(maloaichucdanh) from tmptest) < 9*  
*Begin*  
*Update tmptest set maloaichucdanh maloaichucdanh \* 1.1*  
*Print 'Tang len 10%'*  
*End*
- **Bài tập:** Chương này có tính chất ôn lại về câu lệnh SQL nên yêu cầu sinh viên làm lại các bài tập về các câu lệnh T-SQL đã học trong môn CSDL quan hệ, có áp dụng thêm các phần mở rộng của T-SQL.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.1 View

### 6.1.1 Giới thiệu

- Các View (hay còn gọi là các bảng nhìn) cho phép bạn chia ngang hoặc chia dọc thông tin từ một hoặc nhiều bảng trong một CSDL.
- Ví dụ: Create View test as  
Select ho, ten from canbo



# Chương 6: View, Trigger, Stored procedure và các function

- Về cơ bản các View có các lợi ích thứ nhất là có thể làm đơn giản hoá giao diện người dùng, người dùng lúc này có thể thực hiện những câu truy vấn đơn giản thay vì phải cung cấp các thông số mỗi khi truy vấn được chạy.
- Thứ hai với View bạn có thể thiết lập chế độ an toàn cho phép người dùng được nhìn thấy những gì cần thiết theo ý bạn muốn, nó tối ưu và đơn giản hơn việc bạn phải cấp quyền trên từng cột cho người dùng.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.1.2 Các thao tác

- Tạo View cũng như tạo bảng có hai cách thứ nhất dùng câu lệnh SQL và thứ hai dùng giao diện.
- Dùng câu lệnh cú pháp cơ bản như sau:

*Create View tên\_view as select\_statement.*

- Ví dụ: *Create View test as*

*Select ho, ten, machucdanh from  
canbo,chucdanhcanbo*

*where canbo.macanbo=chucdanhcanbo.macanbo*

# Chương 6: View, Trigger, Stored procedure và các function

- Chỉnh sửa view ta làm như với bảng với lưu ý rằng chỉnh sửa view tức là bạn đang chỉnh sửa trong bảng cơ sở của nó, và không thể chỉnh sửa cùng lúc nhiều hơn một bảng cơ sở.
- Xoá View cũng tương tự như xoá bảng.
- Nếu dùng giao diện bạn bấm chuột phải vào View chọn New View, sau đó bấm chuột phải vào dấu cộng trên thanh công cụ (Add Table), chọn những bảng mà cần dùng để tạo View, bấm add lần lượt từng bảng, cuối cùng bấm close. Từ đây bằng thao tác kéo thả ta có thể thiết kế View theo ý muốn.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.1.3 Các loại và Mục đích sử dụng

- Về cơ bản là có hai loại View, thứ nhất là View projection đơn giản là một tập các cột trong một bảng.
- Thứ hai là View joint bao gồm các cột trong các bảng khác nhau được kết nối lại theo một điều kiện nhất định

## 6.1.4 Ví dụ

- Thứ nhất là View projection  
*Create View test as*  
*Select ho, ten from canbo*

# Chương 6: View, Trigger, Stored procedure và các function

- Thứ hai là View joint

*Create View test as*

*Select ho, ten, machucdanh from  
canbo,chucdanhcanbo*

*where canbo.macanbo=chucdanhcanbo.macanbo*

*Create View test2 as*

*Select kk1.ho, kk1.ten, kk2.que from kk1,kk2*

*where kk1.id=kk2.id*

*Update test2*

*Set ho ='aa',ten='ss', que ='gg' báo lỗi vì cập nhật đến  
nhiều hơn một bảng cơ sở.*

# Chương 6: View, Trigger, Stored procedure và các function

## 6.2 Trigger

### 6.2.1 Giới thiệu

Trigger là một loại thủ tục lưu trữ đặc biệt tự động thực hiện khi bạn hiệu chỉnh, xoá , chèn dữ liệu.

### 6.2.2 Các thao tác

- Cú pháp tạo mới một Trigger như sau:

*Create Trigger tên\_Trigger*

*On table\_name (hoặc View\_name)*

*[For ..Alter | INSTEAD OF] {Insert | Update | Delete}*

*As sql\_statement*

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ:

*Create Trigger test*

*On kk1*

*For Delete as*

*Delete kk2 From deleted where deleted.id = kk2.id*

- Lúc này khi xoá trên bảng kk1, thì Trigger test sẽ tự động được thực hiện và xoá tất cả những record tương ứng trên bảng kk2.

## 6.2.3 Bảo đảm toàn vẹn dữ liệu bằng Trigger

- Về cơ bản Trigger có tác dụng để đảm bảo toàn vẹn dữ liệu, nhưng từ phiên bản 7.0 của SQL Server việc đảm bảo toàn vẹn dữ liệu đã được thực hiện bằng DRI nên Trigger dần ít được sử dụng

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ bây giờ ta muốn thêm một record vào bảng kk1, với điều kiện là id ở bảng kk1 đã phải tồn tại ở bảng kk2 (dạng toàn vẹn tham chiếu tương tự như bảng đơn hàng và bảng khách hàng), ta tạo Trigger như sau:

*Create Trigger hh on kk1*

*for insert as*

*if (Select Count(\*) from kk2,inserted where kk2.id = inserted.id)=0*

*Begin*

*Print 'Khong ton tai id nay'*

*RollBack Transaction*

*End*

- Ngoài ra bạn cũng có thể xem thông tin về Trigger trong một bảng bằng cách chạy thủ tục sau:
- Exec sp\_helptrigger tên\_bảng.



# Chương 6: View, Trigger, Stored procedure và các function

## 6.3 Stored procedure

### 6.3.1 Giới thiệu

- Thủ tục lưu trữ hệ thống được xây dựng sẵn bằng các phát biểu T-SQL, được lưu trữ trong một CSDL của SQL Server.
- Lợi ích cơ bản nhất của Stored procedure là khả năng dùng lại và tốc độ thực hiện, nếu trong một phiên làm việc bạn đã triệu gọi một lần thì nó sẽ được lưu lại ở vùng đệm, nên lần triệu gọi sau nó sẽ được gọi trực tiếp từ vùng đệm.
- Ngoài ra bạn cũng có thể gán quyền cho người sử dụng trên một Stored procedure mà không nhất thiết phải gán quyền trên bảng cơ sở

# Chương 6: View, Trigger, Stored procedure và các function

## 6.2.3 Các thao tác

- Tạo Stored procedure bạn bấm chuột phải vào mục Stored Procedures và chọn new Stored Procedures, soạn thảo lệnh tạo có dạng như sau:

*Create Procedure Tên\_Stored Procedure*

*@parameter(nếu có) Kiểu dữ liệu của tham số*

*As*

*Sql\_Statement*

*Go*

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ: *Create Procedure vidu*  
*as Select \* from chucdanh*

*Go*

- Sau đó chạy thủ tục bằng câu lệnh **Exec vidu** ta sẽ nhận được kết quả là tất cả các cột trên bảng chucdanh
- Bạn cũng có thể tạo một nhóm các thủ tục giống như một bó, phân biệt các thủ tục trong nhóm với thông số **;number**.
- Ví dụ tạo một nhóm thủ tục tên là sp\_vidu gồm ba thủ tục ta làm như sau:

# Chương 6: View, Trigger, Stored procedure và các function

```
Create Procedure sp_vidu;1
```

```
As Select * from chucdanh
```

```
GO
```

```
Create Procedure sp_vidu;2
```

```
As Select * from canbo
```

```
GO
```

```
Create Procedure sp_vidu;3
```

```
As Select * from machucdanh
```

```
GO
```

- Nếu bạn muốn chạy một thủ tục nào trong đó bạn chỉ cần thực hiện

Exec sp\_vidu;2 lúc này ta sẽ được một danh sách tất cả các cột trong bảng canbo.

- Lưu ý khi xóa bạn không thể xóa từng thủ tục một mà phải xóa cả nhóm. 156

# Chương 6: View, Trigger, Stored procedure và các function

## 6.3.3 Sử dụng các thông số và tùy chọn

- Các thông số cho phép bạn tạo các thủ tục lưu trữ và truyền giá trị vào cho nó.
- Lúc này thủ tục lưu trữ hoạt động gần giống như chương trình con trong ngôn ngữ lập trình.

# Chương 6: View, Trigger, Stored procedure và các function

- **Ví dụ:** Ta tạo thủ tục DTB như sau:

Create Procedure DTB

@gt1 smallint,

@gt2 smallint,

@gt3 smallint,

@gt4 smallint,

@gttb smallint output

as

Select @gttb = (@gt1+@gt2+@gt3+@gt4)/4

# Chương 6: View, Trigger, Stored procedure và các function

- Sau đó ta chạy lệnh

*Declare @kk smallint*

*Exec DTB 10,6, 7, 9, @kk output*

*Select 'Gia tri TB la:',@kk*

*Go*

- Ngoài ra bạn cũng có thể đặt thêm tùy chọn With Recompile vào câu lệnh tạo thủ tục hoặc chạy thủ tục, khi đặt tùy chọn này thì mỗi khi thủ tục được triệu gọi thì nó lại được biên dịch lại từ đầu.

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ

*Create Procedure vidu With Recompile  
as Select \* from chucdanh*

*Go*

- Muốn buộc tất cả các thủ tục lưu trữ hay các Trigger mà tham chiếu tới một bảng đặc biệt nào đó phải biên dịch lại ở phần chạy kế tiếp ta dùng thủ tục
- Exec sp\_recompile Tên bảng.



# Chương 6: View, Trigger, Stored procedure và các function

## 6.3. 4 Ví dụ

```
CREATE PROCEDURE kk1_XOA
```

```
    @ma int
```

```
AS
```

```
    delete from kk1 where id=@ma
```

```
GO
```

- Lúc này ta chạy dòng lệnh

```
exec kk1_Xoa 1
```

- Ta đã truyền tham biến là @ma = 1, thủ tục sẽ xoá trong bảng kk1 record có id = 1.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.4 Các Function

- Một tính năng mạnh của SQL Server 2000 là cho phép người dùng tự tạo ra các hàm do chính mình định nghĩa, chứ không nhất thiết phải phụ thuộc vào các hàm mà hệ thống cung cấp. Việc này rất lợi ích khi làm việc với những ứng dụng lớn có những yêu cầu phức tạp mang tính chuyên nghiệp cao.
- Giá trị trả về của các hàm do người dùng định nghĩa có thể là giá trị vô hướng (giá trị đơn) có thể là một bảng.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.4.1 Các hàm vô hướng

- Là các hàm mà trả về giá trị đơn
- Cú pháp cơ bản như sau:

*Create Function Tên hàm(danh sách biến Kiểu dữ liệu tương ứng)*

*Returns Kiểu dữ liệu trả về*

*AS*

*Begin*

*Thân chương trình...*

*Return Giá trị trả về*

*End*

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ 1: Xây dựng một hàm tính tổng của ba số nguyên cho trước.

```
CREATE FUNCTION F_Tong (@x int, @y int, @z int)
```

```
RETURNS int
```

```
AS
```

```
BEGIN
```

```
Declare @gt1 int
```

```
Declare @gt2 int
```

```
Declare @gt3 int
```

```
Declare @tong int
```

```
SET @gt1 =@x
```

```
Set @gt2 = @y
```

```
Set @gt3 = @z
```

```
Set @tong = @gt1+@gt2+@gt3
```

```
Return @tong
```

```
END
```

# Chương 6: View, Trigger, Stored procedure và các function

- Để gọi hàm ta dùng câu lệnh Select Tên hàm(ds biến)
- Như ví dụ trên ta dùng câu lệnh Select dbo.F\_Tong(10, 11, 12) ta sẽ nhận được kết quả là 33.
- Ví dụ 2: Tìm số lớn nhất trong ba số nguyên cho trước.

```
CREATE FUNCTION F_Max (@x int, @y int, @z int)  
RETURNS int  
AS
```

# Chương 6: View, Trigger, Stored procedure và các function

```
BEGIN  
Declare @gt1 int  
Declare @gt2 int  
Declare @gt3 int  
Declare @max int  
Set @gt1 =@x  
Set @gt2 = @y  
Set @gt3 = @z  
Set @max = @gt1  
If @max < @gt2 Set @max = @gt2  
If @max < @gt3 Set @max = @gt3  
Return @max  
END
```

# Chương 6: View, Trigger, Stored procedure và các function

- Lưu ý rằng các câu lệnh và hàm sử dụng trong hàm là các câu lệnh và hàm của SQL.

## 6.4.2 Các hàm Inline Table – Valued

- Tương tự như hàm trên nhưng giá trị trả về của hàm này là một bảng và không có cặp lệnh Begin...end.
- Cú pháp:

*Create Function Tên hàm(danh sách biến Kiểu dữ liệu)*

*Returns Table*

*AS*

*Return Select\_Statement*

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ: Trả về danh sách những cán bộ có những ký tự bắt đầu cho trước.

```
CREATE FUNCTION F_Table1(@bien nvarchar(10))
```

```
RETURNS Table
```

```
AS
```

```
Return (Select * from canbo where ten like @bien+'%')
```

- Để gọi hàm tương tự như trên ta dùng câu lệnh Select.
- Ví dụ ta muốn lấy về tên những cán bộ có chữ cái bắt đầu bằng 't' ta dùng câu lệnh

```
Select ten from dbo.F_Table1('t')
```



# Chương 6: View, Trigger, Stored procedure và các function

## 6.4.3 Các hàm Multistatement Table – Valued

- Tương tự như hàm trên là trả về giá trị là một bảng nhưng ở đây cú pháp có khác và phức tạp hơn, tuy nhiên đổi lại ở đây ta có khả năng tùy biến dữ liệu mạnh hơn

# Chương 6: View, Trigger, Stored procedure và các function

- Cú pháp:

*Create Function Tên hàm(ds biến Kiểu dữ liệu tương ứng)*

*Returns @Tên bảng Table(danh sách các cột và kiểu dữ liệu tương ứng của bảng được trả về)*

*AS*

*Begin*

*Thân chương trình*

*Return [ ( ] select\_statement [ ) ]*

*End*

# Chương 6: View, Trigger, Stored procedure và các function

- Ví dụ:

Viết hàm lấy về một bảng chứa các cột ho, ten trong bảng canbo(macanbo, ho ,ten, ...) và cột tenchucdanhhdd trong bảng chucdanh(machucdanh, tenchucdanhhdd , ...), liên hệ giữa hai bảng này có bảng ChucdanhCanbo(macanbo, machucdanh, idAo).

```
CREATE FUNCTION F_Table2()
```

```
RETURNS @ketqua Table(ho nvarchar(30), ten  
    nvarchar(20), tenChucdanhDD nvarchar(50))
```

```
AS
```

# Chương 6: View, Trigger, Stored procedure và các function

*Begin*

*Declare @temptb Table (ho nvarchar(30), ten nvarchar(20),  
machucdanh smallint)*

*Insert @temptb*

*Select ho, ten, machucdanh from canbo, chucdanhcanbo  
where canbo.macanbo = chucdanhcanbo.macanbo*

*Insert @ketqua*

*Select ho, ten, tENCHUCDANHDD from @temptb tp, chucdanh cd  
where tp.machucdanh=cd.machucdanh*

*Return*

*End*

# Chương 6: View, Trigger, Stored procedure và các function

- Ở đây ta thấy khả năng tùy biến và xử lý dữ liệu cao hơn hẳn hàm ở mục trước, ta khai báo một bảng tạm @temptb chứa 3 cột ho, ten, machudanh.
- Sau đó mới dùng bảng tạm này kết nối với bảng chucdanh để đưa kết quả vào bảng @ketqua của ta.
- Lưu ý khi dùng bảng tạm thì ta phải đặt bí danh cho nó, ví dụ như trên ta đặt bí danh là tp.
- Gọi hàm thì cũng tương tự như trên ta có thể dùng câu lệnh `Select * from dbo.F_Table2()`.

# Chương 6: View, Trigger, Stored procedure và các function

## 6.5 Bài tập

**Bài 1:** Bạn tự thiết kế một CSDL sau đó tạo các Trigger, SP, Function, View theo ý bạn tự nghĩ ra.

**Bài 2:** Bài 2: Có CSDL có ba bảng KháchHang(kh\_id, ten, tuoi, quequan, diachi, sdt), bảng DonHang(dh\_id, kh\_id, ma\_hang, soluong), và bảng Hang(ma\_hang, ten\_hang, gia).

1. Hãy tạo View tên là TongHop chứa các trường (kh\_id, ten, ten\_hang, soluong, gia, tonggia) với nội dung là liệt kê ra tên tất cả các khách hàng đã mua những mặt hàng gì và với số lượng là bao nhiêu, và có tổng giá là bao nhiêu.

# Chương 6: View, Trigger, Stored procedure và các function

2. Hãy tạo một SP với tác dụng như trên và đặt tên là SP\_TongHop, và một SP tên là SP\_TimKiem lấy ra tất cả những khách hàng (gồm kh\_id và tên khách hàng) mà mua hàng với tổng giá trị các mặt hàng lớn hơn một giá trị nhập vào từ bàn phím .
3. Hãy tạo Trigger mà thực hiện toàn vẹn tham chiếu giữa hai bảng (nghĩa là khi thêm mới một bản ghi vào bảng DonHang thì yêu cầu kh\_id phải tồn tại trong bảng KhachHang, và khi xoá khách hàng trong bảng KhachHang thì các bản ghi tương ứng trong bảng DonHang cũng phải tự động được xoá).

# Chương 6: View, Trigger, Stored procedure và các function

4. Xoá các Trigger trên, và tạo ra Trigger trên hai bảng trên sao cho không cho phép người dùng xoá, sửa gì trên hai bảng đó.
5. Hãy tạo ra UDF (hàm tự định nghĩa) tính giá trung bình của những mặt hàng đã được bán, và tạo ra UDF trả về danh sách giá những mặt hàng đã được bán .

**Bài 3:** Bạn hãy tạo ra các Function tính min của bốn số nguyên cho trước, Tính trung bình cộng của 5 số cho trước, ....



# Chương 6: View, Trigger, Stored procedure và các function

**Bài 1:** Tự làm

**Bài 2:**

1. Tạo View TongHop:

- Dùng giao diện sau đó add ba bảng vào và tích vào những trường có trong View, tạo thêm một cột là tonggia = gia\*soluong bằng cách ở cột column gõ gia\*soluong và ở cột Alias gõ tonggia.
- Câu lệnh tương ứng sẽ như sau:

# Chương 6: View, Trigger, Stored procedure và các function

```
CREATE VIEW dbo.TongHop
AS
SELECT    dbo.KhachHang.kh_id, dbo.KhachHang.ten,
          dbo.Hang.ten_hang, dbo.DonHang.soluong, dbo.Hang.gia,
          dbo.Hang.gia * dbo.DonHang.soluong AS tonggia
FROM      dbo.DonHang INNER JOIN
          dbo.Hang ON dbo.DonHang.ma_hang =
          dbo.Hang.ma_hang INNER JOIN
          dbo.KhachHang ON dbo.DonHang.kh_id =
          dbo.KhachHang.kh_id
GO
```

# Chương 6: View, Trigger, Stored procedure và các function

```
2. CREATE PROCEDURE SP_TongHop AS
SELECT    dbo.KhachHang.kh_id, dbo.KhachHang.ten,
          dbo.Hang.ten_hang, dbo.DonHang.soluong, dbo.Hang.gia,
          dbo.Hang.gia * dbo.DonHang.soluong AS tonggia
FROM      dbo.DonHang INNER JOIN
          dbo.Hang ON dbo.DonHang.ma_hang =
          dbo.Hang.ma_hang INNER JOIN
          dbo.KhachHang ON dbo.DonHang.kh_id =
          dbo.KhachHang.kh_id
GO
```

# Chương 6: View, Trigger, Stored procedure và các function

```
CREATE PROCEDURE SP_TimKiem
@bien int
AS
SELECT kh_id,ten,Sum(tonggia) as cost
FROM TongHop
GROUP BY kh_id,ten
HAVING (SUM(tonggia) > @bien)
GO
```

# Chương 6: View, Trigger, Stored procedure và các function

3. *Create Trigger TG\_KiemTra on DonHang*

*for insert as*

*if (Select Count(\*) from KhachHang,inserted where KhachHang.kh\_id  
= inserted.kh\_id)=0*

*Begin*

*Print 'Khong ton tai kh\_id nay'*

*RollBack Transaction*

*End*

*Create Trigger TG\_Xoa*

*On KhachHang*

*For Delete as*

*Delete DonHang From deleted where deleted.kh\_id = DonHang.kh\_id*

# Chương 6: View, Trigger, Stored procedure và các function

## 4. Drop Trigger TG\_KiemTra

*Drop Trigger TG\_Xoa*

*Create Trigger TG\_Cam on KhachHang*

*for delete, update as*

*Begin*

*RollBack Transaction*

*End*

*Create Trigger TG\_Cam on DonHang*

*for delete, update as*

*Begin*

*RollBack Transaction*

*End*

# Chương 6: View, Trigger, Stored procedure và các function

5. *CREATE VIEW* *dbo.View\_temp*

*AS*

```
SELECT    dbo.Hang.ma_hang, dbo.Hang.gia  
FROM      dbo.DonHang INNER JOIN  
            dbo.Hang ON dbo.DonHang.ma_hang =  
            dbo.Hang.ma_hang
```

*Go*

*CREATE FUNCTION F\_GiaTrungbinh ()*

*RETURNS smallmoney*

*AS*

*BEGIN*

*Return (select avg(gia) from View\_temp)*

*END*

# Chương 6: View, Trigger, Stored procedure và các function

```
CREATE FUNCTION F_CacGia()  
RETURNS Table  
AS  
Return (SELECT Distinct(gia)  
FROM View_Temp)
```



# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.1 Sao lưu

### 7.1.1 Giới thiệu và ý nghĩa của việc sao lưu

- Với một thế giới đầy hiểm họa như ngày nay, và một cuộc sống không có gì là không thể xảy ra thì ta luôn luôn phải lường trước hết được mọi việc kể cả việc toàn bộ CSDL của ta bị hỏng.
- Vì vậy việc sao lưu CSDL là vô cùng quan trọng, dù rằng việc này đem lại sự nhàm chán, và đôi khi bị xem là có tính cẩn thận đàn bà. Nhưng với một hệ thống chuyên nghiệp việc sao lưu CSDL luôn là một phần của hệ thống, và thậm chí người quản trị CSDL phải có hẳn một chiến lược sao lưu làm sao cho hợp lý nhất.

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.1.2 Các thao tác

- Lệnh Sao lưu toàn bộ CSDL.
- Bấm chuột phải vào CSDL cần sao lưu, chọn All tasks, chọn tiếp Backup Database. Ở màn hình SQL Server Backup bạn bấm vào Add, lúc này xuất hiện màn hình Select Backup Destination, ở mục File name bạn chọn đường dẫn đến vị trí sẽ chứa file backup CSDL của bạn. Sau đó bấm OK hai lần.

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## Lệnh sao lưu một phần CSDL

- Ta có thể thực hiện điều này khi ta đã có một bản sao lưu toàn bộ CSDL.
- Lúc này ở các lần tiếp theo của việc sao lưu bạn không cần phải sao lưu lại toàn bộ CSDL (nếu CSDL lớn thì việc này chiếm rất nhiều thời gian và công sức) mà chỉ cần sao lưu những thay đổi so với bản sao lưu toàn bộ bằng cách như trên nhưng ở màn hình SQL Server Backup lúc này bạn tích vào Database – Differential.
- Lúc này bạn sẽ có bản sao lưu CSDL tại thời điểm mà bạn tạo backup.
- Bạn có thể đặt một tên khác cho bản sao lưu này hay đơn giản bạn dùng luôn tên của bản sao lưu toàn bộ nhưng hệ thống sẽ tự động phân biệt cho bạn.

# Chương 7: Sao lưu, Khôi phục và Nhân bản

- Ngoài ra bạn cũng có thể sao lưu bằng cách copy các file mdf và ldf của CSDL cần sao lưu vào một thiết bị sao lưu nào đó.
- Lúc khôi phục lại CSDL nếu tên CSDL vẫn đang còn bạn bấm chuột phải vào CSDL đó chọn All task, chọn tiếp Detach Database, bấm OK (nếu như còn một chương trình nào đó đang dùng CSDL này bạn bấm vào Clear).

# Chương 7: Sao lưu, Khôi phục và Nhân bản

- Sau đó bạn bấm chuột phải vào mục Databases, chọn All task, chọn tiếp Attach Database, bấm vào dấu ... chọn đường dẫn đến nơi chứa file mdf của CSDL mà bạn cần khôi phục, bấm OK.
- Lưu ý: Nếu bạn Detach Database thì các file mdf và ldf không mất đi, khác với khi bạn chọn delete CSDL các file mdf và ldf cũng bị xoá luôn

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.1.3 Ví dụ

- Tạo một CSDL tên là test, tạo một bảng tên là test1(ten, tuoi) nhập dữ liệu cho bảng đó. Sau đó thao tác y hệt như trên, ở mục File name đặt tên file là test\_backup.

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.2 Khôi phục

### 7.2.1 Giới thiệu và những trường hợp của khôi phục

- Tất nhiên đã có sao lưu thì phải có quá trình khôi phục nếu không thì sao lưu chẳng mang một ý nghĩa gì, nếu có hỏng hóc của CSDL thì ta có thể khôi phục bất kỳ phiên bản nào mà ta muốn, và khôi phục bất kỳ CSDL nào mà ta muốn (tất nhiên là phải có bản sao lưu của CSDL đó).

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.2.2 Các thao tác

### Khôi phục toàn bộ CSDL:

- Nếu CSDL của bạn bị hỏng hóc bạn muốn sao lưu lại thì bấm chuột phải vào CSDL cần sao lưu chọn All Tasks, chọn tiếp Restore Database, ở màn hình Restore Database tích vào From device, sau đó bấm vào Select Devices.
- Ở màn hình Choose Restore Devices bấm vào Add, ở màn hình Choose Restore Destination ở mục File name ta trở đường dẫn đến file backup của ta sau đó bấm OK liên tiếp là ta đã hoàn tất quá trình sao lưu.<sup>192</sup>



# Chương 7: Sao lưu, Khôi phục và Nhân bản

- Lưu ý: Nếu CSDL của ta bị xoá hẳn, bạn phải tạo mới một CSDL với tên như cũ sau đó lặp lại quá trình trên.
- Và nếu CSDL của bạn đã được Restore một lần thì ở các lần Restore tiếp theo bạn không cần lặp lại quá trình trên mà ở màn hình Restore Database hệ thống sẽ hiện ra bản backup CSDL toàn bộ cuối cùng của bạn và bạn chỉ đơn giản là bấm Ok.

## Khôi phục một phần CSDL:

- Tương tự như trên nhưng ở màn hình Restore Database lúc này sẽ hiện ra một danh sách các lần sao lưu một phần CSDL và bạn chỉ việc tích vào bản sao lưu nào cần khôi phục mà bạn muốn.

# Chương 7: Sao lưu, Khôi phục và Nhân bản

## 7.2.3 Ví dụ

- Như ví dụ trên bạn chỉ cần thêm mới, xoá, sửa các bảng rồi ở các thời điểm đó bạn tạo các bản sao lưu một phần sau đó khôi phục lại bạn sẽ thấy tác dụng của lệnh sao lưu một phần.

## 7.3 Nhân bản

## 7.4 Bài tập

- Bạn tự thực hành tạo ra một CSDL sau đó thay đổi CSDL đó theo ý của bạn và lưu lại các bản sao lưu sao cho bạn có thể thực hiện được các thao tác ở trên như sao lưu toàn bộ CSDL và sao lưu một phần CSDL.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

## 8.1. Khái niệm Application

### 8.1.1. Application có CSDL và Application không có CSDL

Application hay ta còn hiểu là một phần mềm tin học thực hiện một công việc về một lĩnh vực nào đó.

Về cơ bản có các dạng phần mềm sau:

- Phần mềm hệ thống
- Phần mềm thời gian thực
- Phần mềm nghiệp vụ
- Phần mềm khoa học kỹ thuật
- Phần mềm nhúng
- Phần mềm máy tính cá nhân
- Phần mềm trí tuệ nhân tạo
- Phần mềm công cụ

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Các loại phần mềm này có thể có CSDL hoặc có thể không có CSDL, CSDL rất đa dạng có thể là một định dạng đặc biệt nào đó như SQL, Oracle, ..., cũng có thể chỉ đơn giản là một file text.

## 8.1.2. Vai trò, vị trí của HQT SQL Server nằm ở đâu trong Application

- Nếu phần mềm có CSDL thì lúc này Phần mềm = Code + CSDL, CSDL như ta nói có lịch sử phát triển và rất đa dạng, và để tăng tốc độ truy cập từ chương trình vào CSDL và khả năng lưu giữ dữ liệu các định dạng đặc biệt của CSDL ra đời, SQL Server là một trong số những định dạng đó.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

## 8.2 Kiến trúc và mô hình kết nối từ Application vào CSDL

- Ta đã biết CSDL rất đa dạng, chúng có các định dạng đặc biệt khác nhau, từ đơn giản như file text đến các file định dạng đặc biệt như SQL Server, Oracle, ....
- Vì thế cho nên khó khăn lớn nhất khi lập trình truy xuất CSDL đó là cơ chế kết nối, đọc và xử lý dữ liệu từ các file của những HQT CSDL. Chúng có cấu trúc khác nhau và có cách xử lý riêng biệt.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Điểm chung nhất ta đã biết giữa các HQTCSDL đó là ngôn ngữ SQL, nhưng để sử dụng được câu lệnh SQL tác động vào CSDL, lập trình viên cần sử dụng các trình điều khiển do mỗi HQTCSDL cung cấp.
- Nhưng học cách sử dụng các hàm của trình điều khiển là rất khó khăn đối với các lập trình viên.
- Microsoft đã giải quyết vấn đề này bằng cách xây dựng một tập các hàm giao tiếp tổng quát (API) dành cho kết nối và truy xuất dữ liệu, ta gọi đó là cầu nối ODBC.
- Các chương trình Windows muốn truy xuất dữ liệu lúc này chỉ đơn giản là đăng ký trình điều khiển của HQTCSDL mà mình muốn truy xuất với bộ quản lý ODBC.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

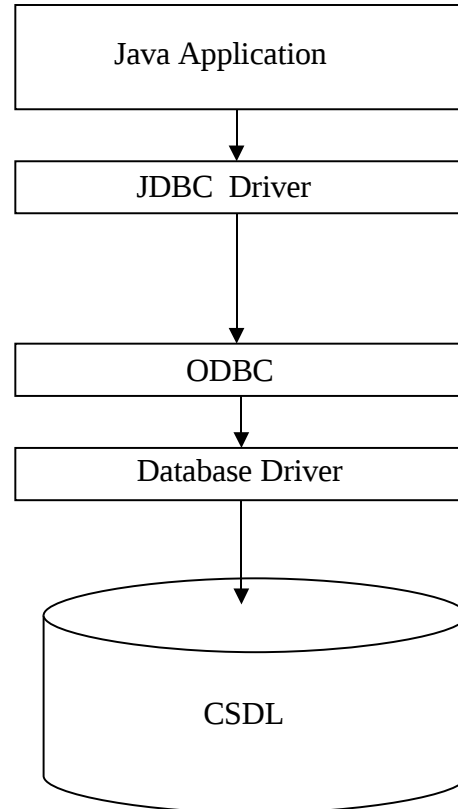
## 8.3. Giới thiệu các loại Ngôn ngữ lập trình và các Mô hình kết nối cụ thể vào HQT CSDL SQL Server

### 8.3.1 Java và mô hình ODBC

- Ngôn ngữ lập trình Java do Sun phát triển, là ngôn ngữ hướng đối tượng, mạnh cho các ứng dụng mạng, cung cấp nhiều công cụ free bởi các hãng thứ ba, được dùng nhiều hiện nay.
- Trình điều khiển JDBC do Java cung cấp, lập trình viên lúc này có khả năng sử dụng JDBC để gọi ODBC và bằng cách này lập trình viên có thể kiểm soát rất nhiều HQT CSDL khác nhau.
- Sun cung cấp sẵn thư viện kết nối JDBC – ODBC kèm theo bộ JDK và dùng miễn phí.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Kiến trúc:





# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Câu lệnh kết nối:

```
String drivername="sun.jdbc.odbc.JdbcOdbcDriver";
```

```
String connectionURL="jdbc:odbc:bkshop";
```

```
String username="";
```

```
String password="";
```

```
Connection con = null;
```

```
Statement stmt = null;
```

```
ResultSet rs = null;
```

```
Class.forName(drivername).newInstance();
```

```
con = DriverManager.getConnection(connectionURL,username,password);
```

```
stmt = con.createStatement();
```

```
String Sql="select * from SellAgent" ;
```

```
rs = stmt.executeQuery(Sql);
```

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Ở đây bkshop là tên ODBC Data source mà bạn đăng ký với ODBC, bước đăng ký như sau:
- Vào Administrative Tools, click đúp chuột vào Data sources (ODBC), chọn mục System DSN, chọn tiếp Add, sau đó chọn trình điều khiển tương ứng với HQTCSDL bạn dùng, ở đây ta dùng SQL Server nên ta chọn SQL Server, click Finish, tiếp theo ở mục Name ta gõ tên ODBC Data source, trong trường hợp này ta gõ là bkshop, ở mục Server ta gõ tên Server của HQTCSDL, ở màn hình tiếp theo bạn phải chọn chế độ đăng nhập vào HQTCSDL

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Ở đây ta có thể chọn chế độ một tài khoản, hoặc chế độ đăng nhập bằng User và password.
- Nếu chọn chế độ một tài khoản thì không cần quan tâm đến user và password, nhưng nếu chọn chế độ đăng nhập bằng User và password thì bạn phải điền đúng user và password.
- Click next ở màn hình tiếp theo bạn tích vào mục *change the default database to* để chọn CSDL mà bạn muốn truy cập tới, Click next liên tục và bấm OK.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Ngoài ra có thể bạn dùng JDBC kết nối trực tiếp với các trình điều khiển của CSDL mà không phải thông qua cầu nối ODBC của Win.
- Tất nhiên lúc này các kết nối này nhanh và hiệu quả hơn so với sử dụng cầu nối ODBC nhưng nhược điểm đó là bạn phải mua các trình điều khiển CSDL đặc thù do những nhà phát triển CSDL hay bên thứ 3 (third – party) cung cấp.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Một cách nữa cũng được dùng nhiều hiện nay đó là JDBC kết nối thông qua các ứng dụng mạng trung gian.
- Theo mô hình lập trình phân tán, ứng dụng trung gian (middle – ware) trên một máy chủ sẽ chịu trách nhiệm kết nối với CSDL.
- Nhiệm vụ của trình khách là sử dụng những trình điều khiển JDBC gọn nhẹ, có khả năng giao tiếp với ứng dụng trung gian qua Internet để truy vấn dữ liệu.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

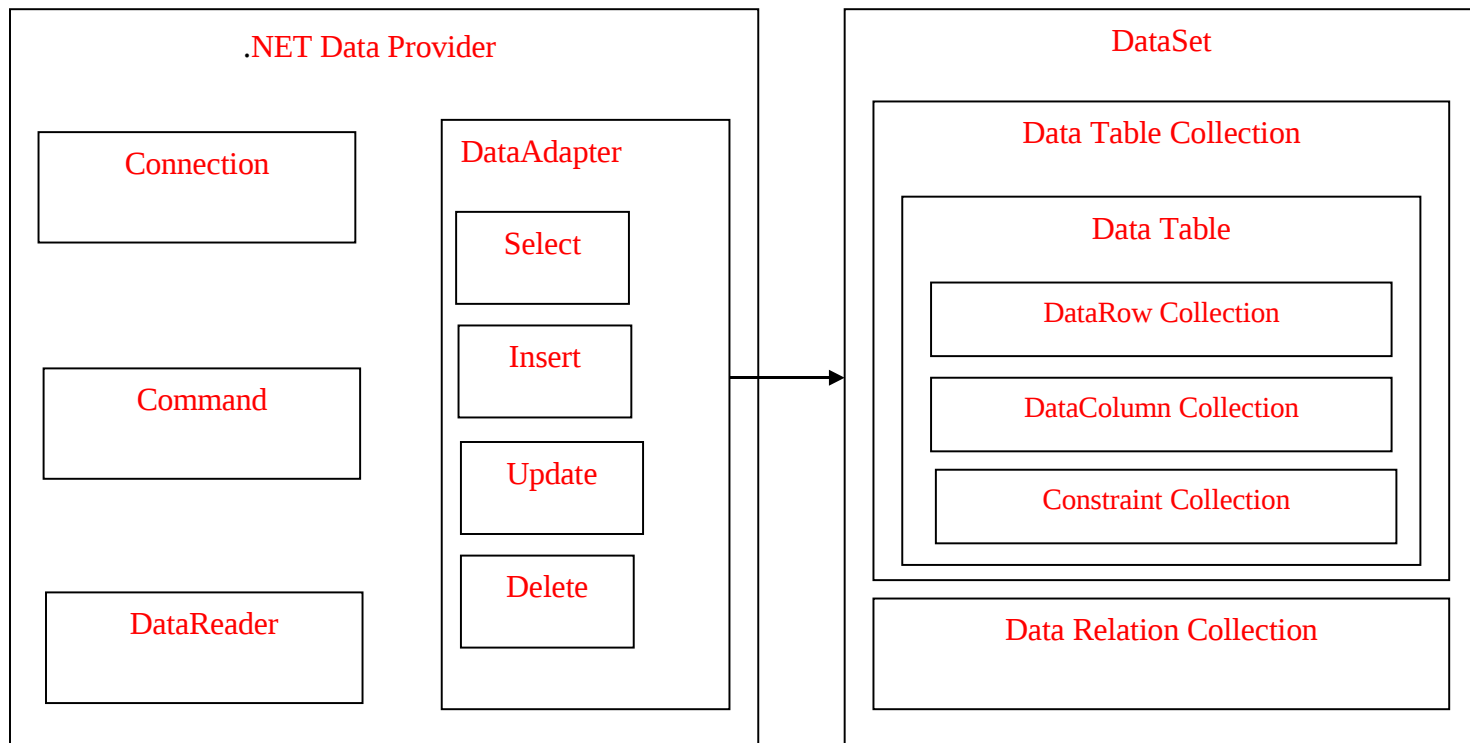
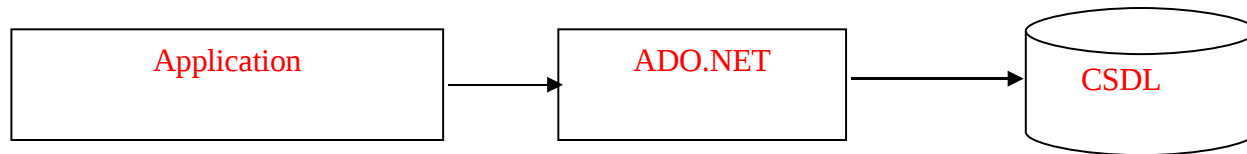
## 8.3.2. .NET và mô hình ADO.NET

- .NET là bộ công cụ, ngôn ngữ lập trình do Microsoft phát triển, có một số tính năng như Java (mục tiêu là để cạnh tranh), có khả năng tích hợp một số ngôn ngữ, không như Java là ngôn ngữ mở, .NET là ngôn ngữ đóng chỉ do mình hãng Microsoft phát triển.
- .NET cũng như Java được dùng nhiều hiện nay.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Kiến trúc tổng quát:
- .NET kết nối với CSDL không cần thông qua cầu nối ODBC, Microsoft đã xây dựng một cầu nối và tập các hàm truy xuất dữ liệu ở dạng tổng quát ở dạng đối tượng (COM) hoạt động gắn với hệ điều hành cho phép truy xuất đến mọi nguồn dữ liệu.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server





# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Câu lệnh kết nối:
- *string constring =*  
*“server=TRINHCUONG;database=DaoTaoHongDuc;uid=sa;pwd=trinh*  
*cuong;”*
- *SqlConnection m\_Conn1 = null;*
- *SqlCommand m\_Cmd;*
- *DataSet m\_DataSet;*
- *DataTable m\_Lophoc;*
- *m\_Cmd = new SqlCommand();*
- *m\_DataSet = new DataSet();*
- *m\_Lophoc = new DataTable();*
- *m\_Conn1 = new SqlConnection(constring );*
- *m\_Conn1.Open();*
- *m\_Cmd.Connection= m\_Conn1;*

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- *m\_Cmd.Parameters.Clear();*
- *m\_Cmd.CommandText= "LOPHOC\_MOLAYCATHEONAM";*
- *m\_Cmd.CommandType= System.Data.CommandType.StoredProcedure;*
- *m\_Cmd.Parameters.Add(new SqlParameter("@namhoc", SqlDbType.SmallInt));*
- *m\_Cmd.Parameters["@namhoc"].Value="2006";*
- *SqlDataAdapter da= new SqlDataAdapter(m\_Cmd);*
  
- *m\_DataSet.Clear();*
- *da.Fill(m\_DataSet, "Lophoc");*
- *m\_Lophoc = m\_DataSet.Tables[0];*
- .....
- *m\_Conn1.Close();*

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Trong đó TRINHCUONG là tên Server của HQTCSDL, DaoTaoHongDuc là tên CSDL, sa và tringcuong lần lượt là username và password của HQTCSDL.
- Khi bạn muốn kết nối và làm việc với CSDL thì trước hết bạn tạo một chuỗi kết nối và một đối tượng Connection để tạo kết nối. Ở trên chuỗi kết nối chính là *constring*, đối tượng kết nối chính là *m\_Conn1*.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Sau đó bạn tạo đối tượng Command thực hiện các câu lệnh SQL tương tác với CSDL, sử dụng đối tượng Command bạn có thể truyền tham số cho câu lệnh SQL nếu bạn sử dụng SP, hoặc bạn có thể sử dụng trực tiếp dòng text không sử dụng SP lúc đó ta thay dòng lệnh `m_Cmd.CommandType = System.Data.CommandType.StoredProcedure;` bằng dòng lệnh `m_Cmd.CommandType = System.Data.CommandType.Text;`

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Nếu là câu lệnh SQL có trả về kết quả (select) thì bạn tạo tiếp một đối tượng DataAdapter, DataSet, DataTable như trên để chứa kết quả trả về. Nếu không thì đơn giản chỉ gọi dòng lệnh `m_Cmd.ExecuteNonQuery();` ngay sau khi truyền xong tham số cho SP.
- Cuối cùng khi thực hiện xong bạn đóng kết nối `m_Conn1.Close();`
- Ở ví dụ trên ta dùng SP tên là `LOPHOC_MOLAYCATHEONAM` và truyền vào một tham số cho SP tên là `@namhoc` có giá trị là 2006.

# Chương 8: Xây dựng phần mềm dùng HQT SQL Server

- Chú ý: Đối với CSDL đặc biệt SQL Server, .NET có hẳn các đối tượng kết nối đặc thù là SqlConnection, SqlCommand, SqlDataAdapter như trong ví dụ trên, các đối tượng này được thiết kế riêng cho CSDL SQL Server nhằm tăng tốc độ và hiệu quả khi làm việc.

## **ĐỀ kiểm tra:**

**Câu 1:** Cơ chế bảo mật trong SQL Server có mấy mức? Hãy trình bày tổng quan từng mức?

**Câu 2:** Bạn hiểu thế nào là quyền phát biểu (Statement Permissions) và quyền về đối tượng (Object Permissions)?

**Câu 3:** Trong CSDL HongDuc, dùng câu lệnh hãy tạo vai trò tên là ĐHTin, sau đó dùng tiếp câu lệnh gán cho vai trò này hai quyền tạo bảng và tạo view.

Giả sử có hai người dùng trong CSDL HongDuc là TinK5 và TinK6

- a. Dùng câu lệnh để gán hai người dùng này là thành viên của vai trò ĐHTin.
- b. Dùng câu lệnh gỡ bỏ quyền tạo bảng của TinK5, nhưng giữ nguyên quyền tạo bảng và tạo view của vai trò ĐHTin, và TinK6 vẫn là thành viên của vai trò ĐHTin.
- c. Xóa vai trò ĐHTin.

**Câu 4:** Giả sử có hai người dùng trong CSDL

HongDuc là nd1 và nd2, hãy dùng câu lệnh cấp cho người dùng nd1 quyền select và update trên bảng Canbo.

Giả sử bảng Canbo có các trường là (Hoten, Tuoi, Gioitinh, Quequan, Chucvu, Luong), dùng câu lệnh cấp cho nd2 quyền select trên các cột (Hoten, Tuoi, Gioitinh, Chucvu), và quyền Update trên các cột (Hoten, Tuoi, Gioitinh) và cho phép nd2 quyền được cấp phép cho các người dùng khác quyền Update này.

Gỡ bỏ quyền Update trên bảng Canbo của nd2 và tất cả những người đã được nd2 cấp cho quyền này.

Vô hiệu hóa quyền select và update trên bảng Canbo của nd1 216