

The background of the slide features a collage of business and technology-related icons and images. On the left, there is a large, glowing cylindrical data storage icon. In the center, a list of business terms is visible: Innovation, Branding, Solution, Marketing, Analysis, Ideas, Success, and Management. To the right, a hand is shown drawing a lightbulb icon, symbolizing an idea or innovation. Below this, there are various other icons including a bar chart, a network diagram, a puzzle piece, and a globe. The overall theme is business and data management.

# Thông minh Quản lý dữ liệu với Máy chủ SQL

Phiên: 6

# Tạo bảng

# Objectives

- Mô tả quy trình tạo, sửa đổi và xóa bảng trong một cơ sở dữ liệu SQL Server
- Mô tả quy trình thêm, sửa đổi và xóa các cột trong bàn

# Creating Tables

Cú pháp:

```
TẠO BẢNG [database_name. [schema_name].| schema_name.]table_name ([<column_name>]  
[data_type] Null/Not Null,)  
BẬT [nhóm tập | "mặc định"]
```

trong

đó, table\_name: là tên của bảng mới, tối đa 128 ký tự. column\_name: là tên của một cột trong bảng. tối đa 128 ký tự. column\_name không được chỉ định cho các cột được tạo bằng kiểu dữ liệu dấu thời gian. Tên cột mặc định của cột dấu thời gian là dấu thời gian.

```
Ví dụ: CREATE TABLE [dbo].[Customer_1](  
    [Customer_id số] [số](10, 0) KHÔNG NULL,  
    [Tên_khách_hàng] [varchar](50) KHÔNG NULL)  
TRÊN [CHÍNH]  
ĐI
```



# Modifying Tables 1-2

Câu lệnh ALTER TABLE được sử dụng để sửa đổi định nghĩa bảng bằng cách thay đổi, thêm hoặc xóa các cột và ràng buộc, chỉ định lại phân vùng hoặc vô hiệu hóa hoặc kích hoạt các ràng buộc và kích hoạt.

Cú pháp:

```
ALTER TABLE [[tên_cơ_sở_dữ_liệu. [tên_lược_đồ].| tên_lược_đồ.]tên_bảng  
ALTER COLUMN ([<tên_cột>] [kiểu_dữ_liệu] Null/Không Null,);  
| XÓA CỘT ([<tên_cột>];  
| ADD ([<tên_cột>] [kiểu_dữ_liệu] Null/Không Null,);
```

Ở đâu,

ALTER COLUMN: chỉ định cột cụ thể nào đó sẽ được thay đổi hoặc sửa đổi.

DROP COLUMN ([<tên\_cột>]: chỉ định rằng tên\_cột sẽ bị xóa khỏi bảng.

ADD: chỉ định rằng một hoặc nhiều định nghĩa cột sẽ được thêm vào.

# Modifying Tables 2-2

Đoạn mã sau minh họa cách thay đổi cột Customer\_id :

```
SỬ DỤNG [CUST_DB]
ALTER TABLE [dbo].[Khách hàng_1]
    ALTER Cột [Số ID khách hàng] [số](12, 0) KHÔNG NULL;
```

Đoạn mã sau minh họa cách thêm cột Contact\_number :

```
SỬ DỤNG [CUST_DB]
ALTER TABLE [dbo].[Bảng_1]
    THÊM [Số_liên_hệ] [số](12, 0) KHÔNG NULL;
```

Đoạn mã sau minh họa việc xóa cột Contact\_number :

```
SỬ DỤNG [CUST_DB]
ALTER TABLE [dbo].[Bảng_1]
    XÓA CỘT [Tên_liên_hệ];
```

Trong một số điều kiện nhất định, các cột không thể bị xóa, chẳng hạn như nếu chúng được sử dụng trong ràng buộc CHECK, FOREIGN KEY, UNIQUE hoặc PRIMARY KEY, được liên kết với định nghĩa DEFAULT, v.v.

# Dropping Tables

Câu lệnh DROP TABLE xóa định nghĩa bảng, dữ liệu của bảng và tất cả các đối tượng liên quan như chỉ mục, kích hoạt, ràng buộc và thông số kỹ thuật về quyền cho bảng đó.

Cú pháp:

```
XÓA BẢNG <Tên_Bảng>
```

Ví dụ:

```
SỬ DỤNG [CUST_DB]  
DROP TABLE [dbo].[Bảng_1]
```

# Column Nullability

Tính năng chấp nhận giá trị null của một cột xác định xem các hàng trong bảng có thể chứa giá trị null cho cột đó hay không.

Giá trị Null không giống với giá trị số không, giá trị trống hoặc chuỗi ký tự có độ dài bằng không (chẳng hạn như ' ').

Tính chất có thể được xác định khi tạo bảng hoặc sửa đổi bảng.

Khi chèn một hàng, nếu không có giá trị nào được đưa ra cho một cột có thể chấp nhận giá trị null, thì SQL Server sẽ tự động đưa ra giá trị null cho hàng đó trừ khi cột đó đã được đưa ra giá trị mặc định.

Ví dụ

```
SỬ DỤNG [CUST_DB]
TẠO BẢNG StoreDetails (StoreID int KHÔNG NULL, Name varchar(40) NULL)
ĐI
```



# DEFAULT Definition

gán giá trị mặc định cho cột nếu không có giá trị nào được cung cấp tại thời điểm tạo.

Khi định nghĩa MẶC ĐỊNH được thêm vào một cột hiện có, SQL Server chỉ áp dụng các giá trị mặc định cho các hàng dữ liệu mới được thêm vào.

Không thể tạo những mục sau trên các cột có định nghĩa MẶC ĐỊNH:

Kiểu dữ liệu dấu thời gian

Thuộc tính IDENTITY hoặc ROWGUIDCOL

Một định nghĩa mặc định hoặc đối tượng mặc định hiện có

Results		Messages	
	ProductID	Name	Price
1	111	Rivets	100.00

Ví dụ

SỬ DỤNG [CUST\_DB]

TẠO BẢNG StoreProduct( ProductID int KHÔNG NULL, Name varchar(40) KHÔNG NULL, tiền gao KHÔNG NULL MẶC ĐỊNH (100))

ĐI

CHÈN VÀO dbo.StoreProduct (ProductID, Tên) GIÁ TRỊ (111, 'Đinh tán')

ĐI



# IDENTITY Property 1-3

Được sử dụng để tạo các cột có thể chứa các giá trị tuần tự được tạo tự động để xác định duy nhất từng hàng trong một bảng.

Thường được sử dụng cho các giá trị khóa chính. Các đặc điểm:

phải được xác định bằng một trong các kiểu dữ liệu sau: thập phân, int, số, smallint, bigint hoặc tinyint.

không cần phải chỉ định giá trị hạt giống và giá trị gia tăng. Nếu không chỉ định, giá trị mặc định là 1 sẽ được sử dụng cho cả hai.

Một bảng không thể có nhiều hơn một cột có thuộc tính IDENTITY .

không được phép có giá trị null và không được chứa định nghĩa hoặc đối tượng MẶC ĐỊNH .

không thể cập nhật giá trị của chúng.

Các giá trị chỉ có thể được chèn rõ ràng vào cột danh tính nếu IDENTITY\_INSERT tùy chọn được bật .

# IDENTITY Property 2-3

Sau khi thuộc tính IDENTITY được thiết lập, việc truy xuất các giá trị của cột định danh có thể được thực hiện bằng cách sử dụng từ khóa IDENTITYCOL với tên bảng trong câu lệnh SELECT.

Để biết liệu một bảng có cột IDENTITY hay không, hàm OBJECTPROPERTY() có thể được sử dụng đã sử dụng.

Để lấy tên của cột IDENTITY trong một bảng, COLUMNPROPERTY chức năng được sử dụng.

Cú pháp:

```
TẠO BẢNG <tên_bảng> (  
tên_cột_kiểu_dữ_liệu [IDENTITY[(giá_trị_hạt_giống, giá_trị_tăng_tăng)]] KHÔNG NULL)
```

Ở đâu,

seed\_value: là giá trị hạt giống để bắt đầu tạo giá trị danh tính.

increment\_value: là giá trị gia tăng mỗi lần tăng.

# IDENTITY Property 3-3

Ví dụ:

```
SỬ DỤNG [CUST_DB]
```

```
ĐI
```

```
TẠO BẢNG HRContactPhone ( Person_ID int IDENTITY(500,1) KHÔNG NULL, MobileNumber bigint  
KHÔNG NULL )
```

```
ĐI
```

```
CHÈN VÀO HRContactPhone GIÁ TRỊ(983452201)
```

```
CHÈN VÀO HRContactPhone GIÁ TRỊ(993026654)
```

```
ĐI
```

Hình sau đây cho thấy đầu ra

Results			Messages	
	Person_ID	MobileNumber		
1	500	983452201		
2	501	993026654		

# Globally Unique Identifiers 1-3

Chỉ có thể tạo một cột định danh và một cột định danh duy nhất toàn cầu cho mỗi bảng.

Để tạo và làm việc với các mã định danh duy nhất toàn cầu, cần sử dụng kết hợp ROWGUIDCOL, kiểu dữ liệu `uniqueidentifier` và hàm `NEWID`.

Giá trị cho một cột duy nhất toàn cầu không được tự động tạo ra.

Người ta phải tạo một định nghĩa MẶC ĐỊNH với hàm `NEWID()` cho một cột `uniqueidentifier` để tạo ra một giá trị duy nhất toàn cầu.



# Globally Unique Identifiers 2-3

Hàm `NEWID()` tạo ra một số định danh duy nhất là chuỗi nhị phân 16 byte.

Cột có thể được tham chiếu trong danh sách `SELECT` bằng cách sử dụng từ khóa `ROWGUIDCOL` .

Để biết liệu một bảng có cột `ROWGUIDCOL` hay không , hàm `OBJECTPROPERTY` được sử dụng.

Hàm `COLUMNPROPERTY` được sử dụng để lấy tên của cột `ROWGUIDCOL` .

# Globally Unique Identifiers 3-3

Ví dụ

```
SỬ DỤNG [CUST_DB]
```

```
TẠO BẢNG EMP_CellularPhone( Person_ID uniqueidentifier MẶC ĐỊNH NEWID() KHÔNG NULL,  
PersonName varchar(60) KHÔNG NULL)
```

ĐI

Đoạn mã sau thêm giá trị vào cột PersonName :

```
SỬ DỤNG [CUST_DB]
```

```
CHÈN VÀO EMP_CellularPhone(PersonName) VALUES ('William Smith')
```

```
LỰA CHỌN * TỪ EMP_CellularPhone
```

ĐI

Hình sau đây cho thấy đầu ra trong đó một mã định danh duy nhất được hiển thị so với Tên người cụ thể :

Results		Messages
	Person_ID	PersonName
1	362C4377-D194-4607-A466-7FF02064EAFD	William Smith

# Constraints

Ràng buộc là một thuộc tính được gán cho một cột hoặc một tập hợp các cột trong một bảng để ngăn chặn một số loại giá trị dữ liệu không nhất quán được nhập vào.

được sử dụng để áp dụng các quy tắc logic kinh doanh và thực thi tính toàn vẹn của dữ liệu.

có thể được tạo khi bảng được tạo hoặc thêm vào ở giai đoạn sau

có thể được phân loại thành ràng buộc cột và ràng buộc bảng.

Ràng buộc cột được chỉ định như một phần của định nghĩa cột và chỉ áp dụng cho cột đó.

Ràng buộc bảng có thể áp dụng cho nhiều cột trong một bảng và được khai báo độc lập với định nghĩa cột.

Ràng buộc bảng phải được sử dụng khi có nhiều hơn một cột được bao gồm trong một ràng buộc.

SQL Server hỗ trợ các loại ràng buộc sau:

- KHÓA CHÍNH
- KIỂM TRA
- ĐỘC NHẤT
- KHÔNG NULL
- KHÓA NGOẠI

# PRIMARY KEY

Một bảng thường có khóa chính bao gồm một cột hoặc tổ hợp các cột để xác định duy nhất từng hàng trong bảng.

Ràng buộc PRIMARY KEY được sử dụng để tạo khóa chính và đảm bảo tính toàn vẹn của thực thể trong bảng.

Chỉ có thể tạo một ràng buộc khóa chính cho mỗi bảng.

Cột là khóa chính không thể có giá trị NULL.

Cú pháp:

```
TẠO BẢNG <tên_bảng> (Tên_cột kiểu dữ liệu KHÓA CHÍNH [  
[cột_danh_sách])
```

```
TẠO BẢNG <tên_bảng> (<tên_cột> <kiểu_dữ_liệu> [, danh_sách_cột]  
CONSTRAINT constraints_name KHÓA CHÍNH (cột,..))
```



# UNIQUE

Một ràng buộc DUY NHẤT được sử dụng để đảm bảo rằng chỉ những giá trị duy nhất được nhập vào cột hoặc tập hợp các cột. Ràng

buộc UNIQUE cho phép các giá trị null. Một bảng

đơn có thể có nhiều hơn một ràng buộc UNIQUE.

## Cú pháp:

```
TẠO BẢNG <tên_bảng> ([<danh_sách_cột> <tên_cột> <kiểu_dữ_liệu> DUY NHẤT [<danh_sách_cột>])
```

## Ví dụ:

```
SỬ DỤNG [CUST_DB]
```

ĐI

```
TẠO BẢNG EMP_ContactPhone(Person_ID int PRIMARY KEY, MobileNumber bigint  
DUY NHẤT, Nhà cung cấp dịch vụ varchar(30), Số điện thoại cố định bigint DUY NHẤT)
```

# FOREIGN KEY

Khóa ngoại trong một bảng là một cột trỏ tới một cột khóa chính trong một bảng khác.

Ràng buộc khóa ngoại được sử dụng để thực thi tính toàn vẹn tham chiếu.

Cú pháp:

```
TẠO BẢNG <tên_bảng1>([danh_sách_cột,] <tên_cột> <kiểu_dữ_liệu>  
THAM CHIẾU KHÓA NGOẠI <table_name> (pk_column_name> [, column_list])
```

Ví dụ

```
SỬ DỤNG [CUST_DB]
```

```
ĐI
```

```
TẠO BẢNG EMP_PhoneExpenses ( Expense_ID int KHÓA CHÍNH, MobileNumber  
bigint THAM CHIẾU KHÓA NGOẠI EMP_ContactPhone (MobileNumber), Số tiền bigint)
```

# CHECK

Ràng buộc CHECK giới hạn các giá trị có thể được đặt trong một cột.

Ràng buộc kiểm tra thực thi tính toàn vẹn của dữ

liệu. Ràng buộc CHECK hoạt động bằng cách chỉ định một điều kiện tìm kiếm, có thể đánh giá là ĐÚNG, SAI hoặc chưa biết.

Các giá trị được đánh giá là FALSE sẽ bị từ chối. Có thể chỉ định nhiều ràng buộc CHECK cho một cột. Một ràng buộc CHECK cũng có thể được áp dụng cho nhiều cột bằng cách tạo nó ở cấp độ bảng.

Ví dụ:

```
SỬ DỤNG [CUST_DB]
TẠO BẢNG EMP_PhoneExpenses ( Expense_ID int KHÓA CHÍNH,
  MobileNumber bigint THAM CHIẾU KHÓA NGOẠI EMP_ContactPhone
  (MobileNumber), Số tiền bigint CHECK (Số tiền >10))
ĐI
```

# NOT NULL

Ràng buộc NOT NULL yêu cầu cột không chấp nhận giá trị null.

Ràng buộc NOT NULL được sử dụng để thực thi tính toàn vẹn của miền, tương tự như ràng buộc CHECK .



# Data Modification Statements 1-3

Được sử dụng để sửa đổi dữ liệu, chúng là các câu lệnh INSERT, UPDATE và DELETE.

## Câu lệnh INSERT

thêm một hàng mới vào bảng.

Cú pháp:

```
CHÈN [TRONG] <Tên_bảng>  
GIÁ TRỊ <giá trị>
```

Ví dụ

```
SỬ DỤNG [CUST_DB]  
CHÈN VÀO [dbo].[Bảng_2] GIÁ TRỊ (101, 'Richard Parker', 'Richy')  
ĐI
```

# Data Modification Statements 2-3

## Tuyên bố CẬP NHẬT

Câu lệnh UPDATE sửa đổi dữ liệu trong bảng.

### Cú pháp:

```
CẬP NHẬT <Tên_Bảng>  
ĐẶT <Tên_cột = Giá trị>  
[WHERE <Điều kiện tìm kiếm>]
```

Ở đây,

<Column\_Name>: tên của cột mà bản ghi sẽ được cập nhật.

<Giá trị>: chỉ định giá trị mới cho cột đã sửa đổi.

<Điều kiện tìm kiếm>: điều kiện cần đáp ứng để các hàng được cập nhật.

Ví dụ:

```
SỬ DỤNG [CUST_DB]  
CẬP NHẬT [dbo].[Bảng_2] ĐẶT Contact_number = 5432679 NƠI  
Contact_name GIỐNG NHƯ 'Richy'  
ĐI
```

# Data Modification Statements 3-3

## Câu lệnh DELETE

Câu lệnh DELETE xóa các hàng khỏi bảng. Cú pháp của câu lệnh DELETE như sau:

### Cú pháp:

```
XÓA TỪ <Table_Name>  
[WHERE <Điều kiện tìm kiếm>]
```

Trong

đó, mệnh đề WHERE được sử dụng để chỉ định điều kiện. Nếu mệnh đề WHERE không được bao gồm trong câu lệnh DELETE, tất cả các bản ghi trong bảng sẽ bị xóa.

Ví dụ:

```
SỬ DỤNG [CUST_DB]  
XÓA TỪ [dbo].[Customer_2] NƠI Contact_number = 5432679 ĐI
```



# Summary

- Hầu hết các bảng đều có khóa chính, bao gồm một hoặc nhiều cột trong bảng dùng để xác định duy nhất các bản ghi.
- Tính năng có thể chấp nhận giá trị null của một cột xác định xem các hàng trong bảng có thể chứa giá trị null cho cột đó.
- Một định nghĩa MẠC ĐỊNH cho một cột có thể được tạo tại thời điểm bảng tạo hoặc thêm vào một bảng hiện có ở giai đoạn sau.
- Thuộc tính IDENTITY của SQL Server được sử dụng để tạo các cột định danh có thể chứa các giá trị tuần tự được tạo tự động để xác định duy nhất từng hàng trong bảng.
- Các ràng buộc được sử dụng để áp dụng các quy tắc logic kinh doanh và thực thi dữ liệu chính trực.
- Một ràng buộc DUY NHẤT được sử dụng để đảm bảo rằng chỉ có các giá trị duy nhất được nhập vào một cột hoặc một tập hợp các cột.
- Khóa ngoại trong bảng là một cột trỏ tới khóa chính cột trong bảng khác.
- Ràng buộc CHECK giới hạn các giá trị có thể được đặt trong một cột.