



# Dữ liệu thông minh Quản lý với SQL Máy chủ

Phiên: 10

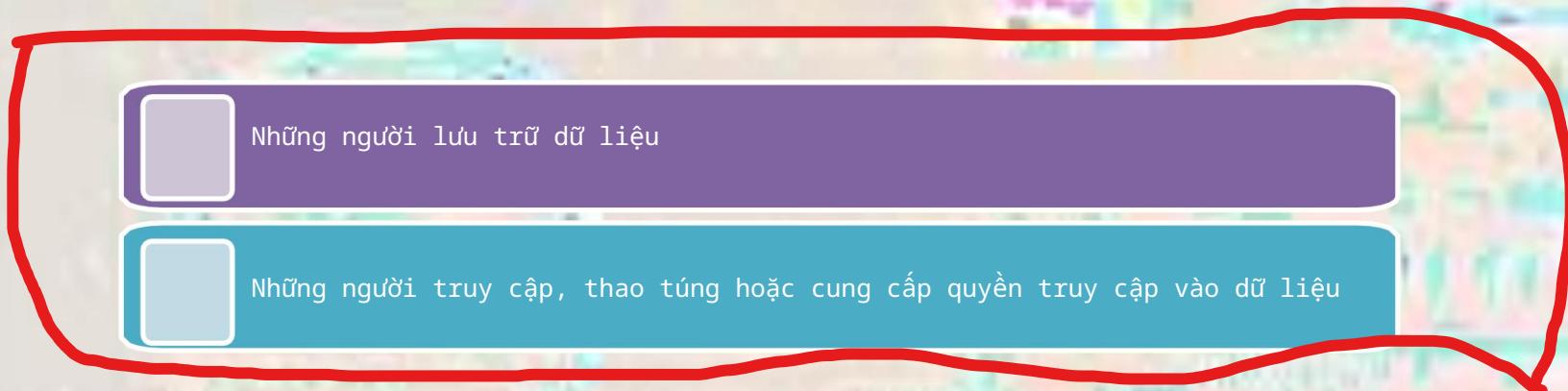
Lượt xem, Quy trình được lưu  
trữ và Truy vấn siêu dữ liệu

# Mục tiêu

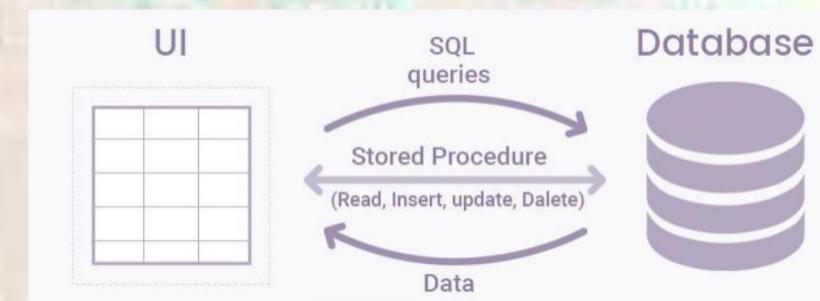
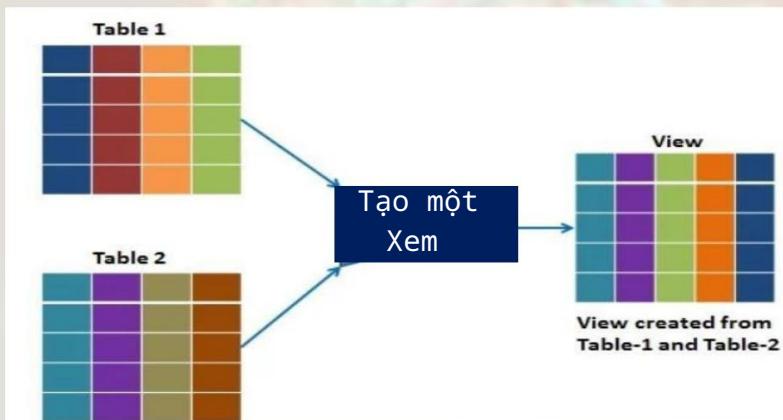
- Xác định chế độ xem
- Mô tả kỹ thuật tạo, thay đổi và xóa chế độ xem
- Xác định các thủ tục được lưu trữ và các loại của nó
- Mô tả quy trình tạo, thay đổi và thực thi các tệp được lưu trữ thủ tục
- Mô tả các thủ tục lưu trữ lồng nhau
- Mô tả truy vấn siêu dữ liệu SQL Server System Catalog quan điểm và chức năng

# Giới thiệu

Cơ sở dữ liệu SQL Server có hai loại đối tượng chính:



Các chế độ xem và thủ tục được lưu trữ thuộc về loại sau.



# Lượt xem views



# Tạo chế độ xem

Người dùng chỉ có thể tạo chế độ xem bằng các cột từ bảng hoặc chế độ xem khác nếu người dùng có quyền truy cập vào các bảng và chế độ xem này.

Chế độ xem được tạo bằng câu lệnh CREATE VIEW và chỉ có thể được tạo trong cơ sở dữ liệu hiện tại.

SQL Server xác minh sự tồn tại của các đối tượng được tham chiếu trong định nghĩa chế độ xem.

Cú pháp:

```
TẠO VIEW <view_name>
AS <câu lệnh chọn>
```

Ví dụ

**TẠO VIEW vwProductInfo AS**

CHỌN ProductID,

ProductName,

SafetyStockLevel

TỪ Sản xuất.Sản phẩm;

ĐI

CHỌN \* TỪ vwProductInfo

ĐI

	ProductID	ProductNumber	Name	SafetyStockLevel
1	1	AR-5381	Adjustable Race	1000
2	2	BA-8327	Bearing Ball	1000
3	3	BE-2349	BB Ball Bearing	800
4	4	BE-2908	Headset Ball Bearings	800
5	316	BL-2036	Blade	800
6	317	CA-5965	LL Crankarm	500
7	318	CA-6738	ML Crankarm	500
8	319	CA-7457	HL Crankarm	500
9	320	CB-2903	Chainring Bolts	1000
10	321	CN-6137	Chainring Nut	1000

Bản ghi từ aView

# Tạo View Sử dụng Từ khóa JOIN 1-3

Từ khóa JOIN cũng có thể được sử dụng để tạo chế độ xem.

Câu lệnh CREATE VIEW được sử dụng cùng với từ khóa JOIN để tạo chế độ xem bằng các cột từ nhiều bảng.

Cú pháp:

```
CREATEVIEW<view_name> AS  
SELECT * FROMtable_name1 JOIN  
table_name2  
ONtable_name1.column_name=table_name2.column_name
```

Ở đâu,

view\_name: chỉ định tên của chế độ xem.

table\_name1: chỉ định tên của bảng đầu tiên.

JOIN: chỉ định rằng hai bảng được nối lại bằng từ khóa JOIN.

table\_name2: chỉ định tên của bảng thứ hai.

# Tạo View Sử dụng Từ khóa JOIN 2-3

## Code Snippet 3:

```
CREATE VIEW vwPersonDetails AS
SELECT
    p.Title
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,e.[JobTitle]
FROM [HumanResources].[Employee] e
    INNER JOIN [Person].[Person] p
        ON p.[BusinessEntityID] = e.[BusinessEntityID]
GO
```

```
SELECT * FROM vwPersonDetails
```

	Title	FirstName	MiddleName	LastName	Job Title
1	NULL	Ken	J	Sánchez	Chief Executive Officer
2	NULL	Temi	Lee	Duffy	Vice President of Engineering
3	NULL	Roberto	NULL	Tamburello	Engineering Manager
4	NULL	Rob	NULL	Walters	Senior Tool Designer
5	Ms.	Gail	A	Erickson	Design Engineer
6	Mr.	Jossef	H	Goldberg	Design Engineer
7	NULL	Dylan	A	Miller	Research and Development Manager
8	NULL	Diane	L	Margheim	Research and Development Engineer
9	NULL	Gigi	N	Matthew	Research and Development Engineer
10	NULL	Michael	NULL	Raheem	Research and Development Manager

# Tạo View Sử dụng Từ khóa JOIN 3-3

Để thay thế tất cả các giá trị NULL trong đầu ra bằng một chuỗi null, hàm COALESCE()

```
CREATE VIEW vwPersonDetailsNew
AS
SELECT
    COALESCE(p.Title, '') AS Title
    ,p.[FirstName]
    ,COALESCE(p.MiddleName, '') AS MiddleName
    ,p.[LastName]
    ,e.[JobTitle]
FROM [HumanResources].[Employee] e
    INNER JOIN [Person].[Person] p
        ON p.BusinessEntityID = e.BusinessEntityID
GO
```

	Title	FirstName	MiddleName	LastName	JobTitle
1		Ken	J	Sánchez	Chief Executive Officer
2		Temi	Lee	Duffy	Vice President of Engineering
3		Roberto		Tamburello	Engineering Manager
4		Rob		Walters	Senior Tool Designer
5	Ms.	Gail	A	Erickson	Design Engineer
6	Mr.	Jossef	H	Goldberg	Design Engineer
7		Dylan	A	Miller	Research and Development Manager
8		Diane	L	Margheim	Research and Development Engineer
9		Gigi	N	Matthew	Research and Development Engineer
10		Michael		Raheem	Research and Development Manager

# Hướng dẫn và hạn chế về quan điểm

Một chế độ xem chỉ được tạo trong cơ sở dữ liệu hiện tại. Các bảng cơ sở và chế độ xem mà chế độ xem được tạo ra có thể từ các cơ sở dữ liệu hoặc máy chủ khác.

Tên chế độ xem phải là duy nhất và không được giống với tên bảng trong lược đồ.

Không thể tạo chế độ xem trên các bảng tạm thời.

Một chế độ xem không thể có chỉ mục toàn văn.

Một chế độ xem không thể chứa định nghĩa MẶC ĐỊNH.

Câu lệnh CREATE VIEW chỉ có thể bao gồm mệnh đề ORDER BY nếu từ khóa TOP được sử dụng.

Chế độ xem không thể tham chiếu tới hơn 1.024 cột.

Câu lệnh CREATE VIEW không thể bao gồm từ khóa INTO.

Câu lệnh CREATE VIEW không thể kết hợp với các câu lệnh Transact-SQL khác trong một đợt duy nhất.

# Sửa đổi dữ liệu thông qua chế độ xem

Có thể sử dụng chế độ xem để sửa đổi dữ liệu trong các bảng cơ sở dữ liệu.

Dữ liệu có thể được chèn, sửa đổi hoặc xóa thông qua chế độ xem bằng cách sử dụng các câu lệnh sau:

CHÈN

CẬP NHẬT

XÓA BỎ

# CHÈN với Views

Câu lệnh INSERT được sử dụng để thêm một hàng mới vào một bảng hoặc một xem.

Trong quá trình thực hiện câu lệnh, nếu giá trị của một cột là không được cung cấp, SQL Server Database Engine phải cung cấp giá trị dựa trên định nghĩa của cột.

Giá trị cho cột được cung cấp tự động nếu cột:

Has an IDENTITY  
property

Has a default value  
specified

Has a timestamp  
data type

Takes null values

Is a computed  
column

# CHÈN với Views

- Ví dụ

SỬ DỤNG StrongHold

```
TẠO BẢNG Employee_Personal_Details (
    EmpID int KHÔNG NULL,
    Tên varchar(30) KHÔNG NULL,
    Họ varchar(30) KHÔNG NULL,
    Địa chỉ varchar(30) NULL )
```

```
TẠO BẢNG Employee_Salary_Details (
    EmpID int KHÔNG NULL,
    Chỉ định varchar(30),
    Lương int KHÔNG NULL)
```

TẠO VIEW vwEmployee\_Personal\_Details

BẢNG

```
CHỌN e1.EmpID, FirstName, LastName, Chức danh, Lương TỪ Employee_Personal_Details e1
```

```
THAM GIA Employee_Salary_Details e2 TRÊN e1.EmpID
```

```
= e2.EmpID TẠO VIEW vwEmpDetails DƯỚI DẠNG CHỌN
```

```
* TỪ Employee_Personal_Details
```

CHÈN VÀO vwEmpDetails GIÁ TRỊ

```
( 1 , 'Jack' , 'NYC' , '123 Florida' )
```

# CẬP NHẬT với Views

Câu lệnh UPDATE có thể được sử dụng để thay đổi dữ liệu trong chế độ xem.

Cập nhật chế độ xem cũng cập nhật bảng bên dưới.

Không thể cập nhật giá trị của cột có thuộc tính IDENTITY.

Không thể cập nhật bản ghi nếu bảng cơ sở chứa cột TIMESTAMP.

Khi có lệnh tự nối với cùng một chế độ xem hoặc bảng cơ sở, câu lệnh UPDATE sẽ không hoạt động.

Trong khi cập nhật một hàng, nếu ràng buộc hoặc quy tắc bị vi phạm, câu lệnh sẽ bị chấm dứt, lỗi sẽ được trả về và không có bản ghi nào được cập nhật.

# CẬP NHẬT với Views

- Ví dụ

SỬ DỤNG StrongHold

TẠO BẢNG Product\_Details (

Mã sản phẩm int,

Tên sản phẩm varchar(30),

Tỷ giá tiền

tệ )

CHÈN VÀO Product\_Details

GIÁ TRỊ ('1','DVD Writer',2250),

('2','Ổ cứng ngoài ',4250)

TẠO VIEW vwProduct\_Details

BẢNG

CHỌN Tên sản phẩm, Tỷ lệ

TỪ Chi tiết sản phẩm

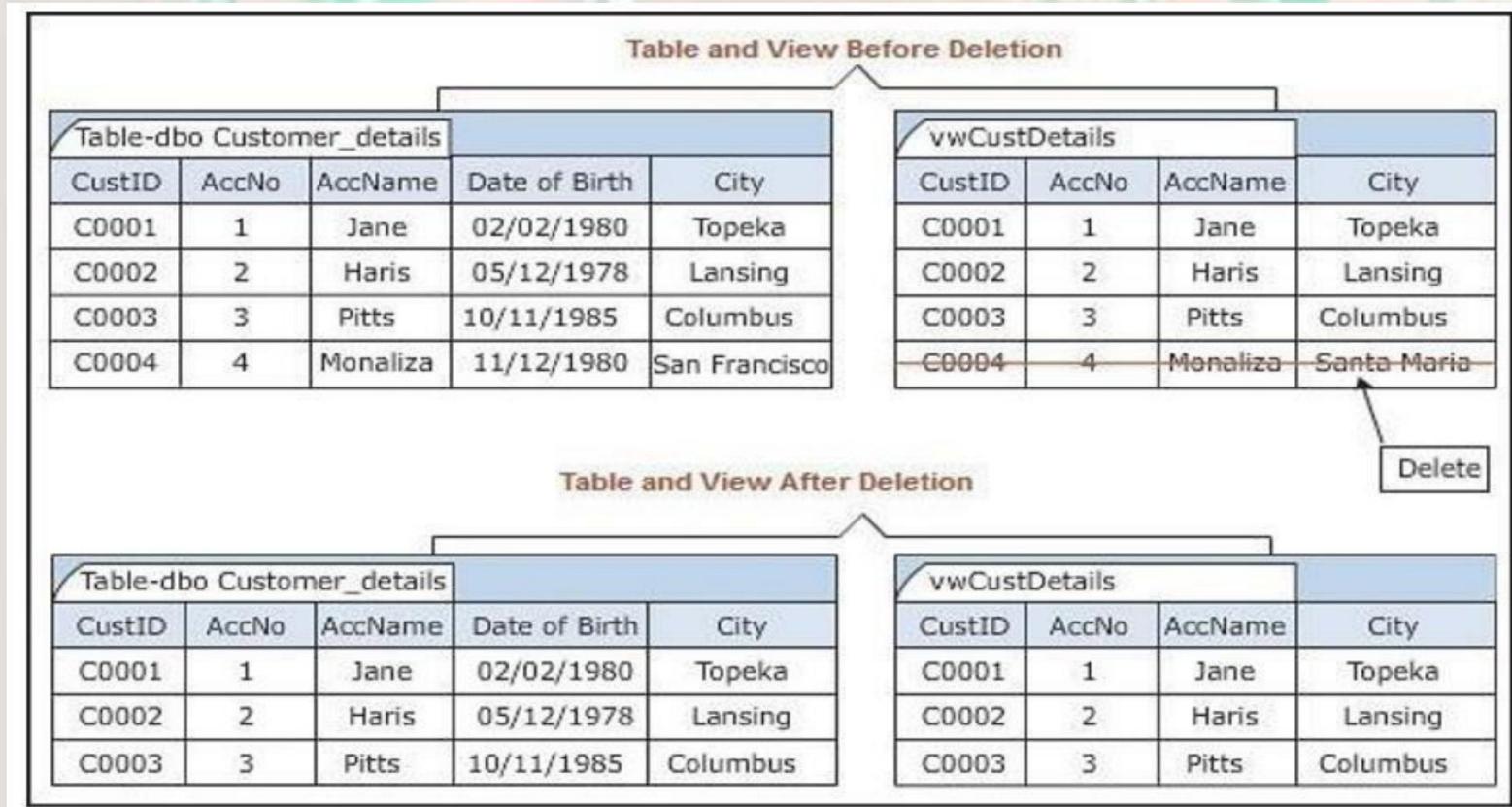
CẬP NHẬT vwProduct\_Details

Tỷ lệ ĐẶT = 3000

NƠI Tên sản phẩm = 'Trình ghi DVD'

# XÓA VỚI CHẾ ĐỘ XEM

SQL Server cho phép bạn xóa các hàng khỏi chế độ xem. Các hàng có thể bị xóa khỏi chế độ xem bằng cách sử dụng câu lệnh DELETE. Khi các hàng bị xóa khỏi chế độ xem, các hàng tương ứng cũng bị xóa khỏi bảng cơ sở.



Xóa khỏi Views

# XÓA với chế độ xem

- Ví dụ

SỬ DỤNG StrongHold

ĐI

TẠO VIEW vwProductDetails

BẰNG

CHỌN ProductID

TỪ Chi tiết sản phẩm

NƠI ProductID=1

ĐI

XÓA TỪ vwProductDetails

ĐI

LỰA CHỌN \*

TỪ Chi tiết sản phẩm

Xóa khỏiViews

# Thay đổi quan điểm

Có thể sửa đổi hoặc thay đổi chế độ xem bằng cách xóa và tạo lại chế độ xem đó hoặc thực thi câu lệnh ALTER VIEW.

ALTER VIEW có thể được áp dụng cho các chế độ xem được lập chỉ mục; tuy nhiên, nó sẽ xóa bỏ vô điều kiện tất cả các chỉ mục trên chế độ xem đó.

Chế độ xem thường bị thay đổi khi người dùng yêu cầu thông tin bổ sung hoặc thực hiện thay đổi trong định nghĩa bảng cơ sở.

# Thay đổi quan điểm

- Cú pháp:

```
ALTER VIEW <tên_lượt_xem>
AS <câu_lệnh_chọn>
```

- Ví dụ

```
THAY ĐỔI XEM vwProduct_Details
BẰNG
CHỌN ProductID,ProductName, Rate
TỪ Chi tiết sản phẩm

ĐI

LỰA CHỌN *
TỪ vwProduct_Details
ĐI
```

# Lượt xem giảm

Một chế độ xem có thể bị xóa khỏi cơ sở dữ liệu nếu nó không còn cần thiết nữa. Điều này được thực hiện bằng cách sử dụng câu lệnh DROP VIEW.

Định nghĩa về chế độ xem và thông tin khác liên quan đến chế độ xem sẽ bị xóa khỏi danh mục hệ thống.

Cú pháp:

XÓA XEM <view\_name>

Ví dụ

XEM THẢ vwProduct\_Details

# Định nghĩa của một View

Định nghĩa của chế độ xem giúp hiểu cách dữ liệu của nó được lấy từ các bảng nguồn.

Cú pháp:

```
sp_helptext <view_name>
```

Ví dụ:

```
EXEC sp_helptext vwProductPrice
```

Text	
1	CREATE VIEW vwEmployee_Personal_Details AS
2	SELECT e1.EmpID, FirstName, LastName, Designatio...
3	JOIN Employee_Salary_Details e2
4	ON e1.EmpID = e2.EmpID

Sử dụng sp\_helptext để hiển thị ViewDefinitions

# Tạo aView bằng cách sử dụng các hàm tích hợp

Có thể tạo chế độ xem bằng các hàm tích hợp của SQL Server.

Khi sử dụng các hàm, cột được dẫn xuất phải bao gồm tên cột trong câu lệnh CREATE VIEW.

TẠO VIEW vwProduct\_Details AS  
CHỌN Tên sản phẩm, AVG(Tỷ lệ) NHƯ  
Tỷ lệ trung bình  
TỪ Chi tiết sản phẩm  
NHÓM THEO Tên sản phẩm  
ĐI

LỰA CHỌN \*  
TỪ vwProduct\_Details  
ĐI

	ProductName	AverageRate
1	Hard Disk Drive	3570.00
2	Portable Hard Drive	5580.00

Sử dụng các hàm tích hợp với Views

# KIỂM TRA TÙY CHỌN

CHECK OPTION là tùy chọn liên quan đến câu lệnh CREATE VIEW.

với

Nó được sử dụng để đảm bảo rằng tất cả các bản cập nhật trong chế độ xem đều đáp ứng các điều kiện được đề cập trong định nghĩa chế độ xem.

Nếu các điều kiện không được đáp ứng, công cụ cơ sở dữ liệu sẽ trả về lỗi.

Cú pháp:

```
CREATE VIEW<view_name>
AS select _statement [ WITH CHECK OPTION ]
```

```
TẠO VIEW vwProductInfo AS
CHỌN * TỪ Production.Product
NƠI SafetyStockLevel <=1000
VỚI TÙY CHỌN KIỂM TRA;
ĐỊ
```

```
CẬP NHẬT vwProductInfo ĐẶT SafetyStockLevel = 2500
NƠI ProductID=321
```

LỖI !

# KIỂM TRA TÙY CHỌN

- Ví dụ

```
TẠO VIEW vwProductInfo AS  
CHỌN * TỪ Product_Details  
Ở ĐÂU Tỷ lệ <= 4000  
VỚI TÙY CHỌN KIỂM TRA  
ĐI  
  
CẬP NHẬT vwProductInfo ĐẶT Tỷ lệ =  
10000  
NƠI ProductID=3
```

# Tùy chọn SCHEMABINDING

Một chế độ xem có thể được liên kết với lược đồ của bảng cơ sở bằng cách sử dụng **tùy chọn SCHEMABINDING**.

Tùy chọn này có thể được sử dụng với các câu lệnh CREATE VIEW hoặc ALTER VIEW.

Khi tùy chọn SCHEMABINDING được chỉ định, các bảng cơ sở không thể được sửa đổi nếu điều đó ảnh hưởng đến định nghĩa chế độ xem.

Định nghĩa chế độ xem phải được sửa đổi hoặc xóa trước để loại bỏ các phụ thuộc vào bảng cần sửa đổi.

TẠO VIEW vwNewProductInfo

VỚI SCHEMABINDING NHƯ

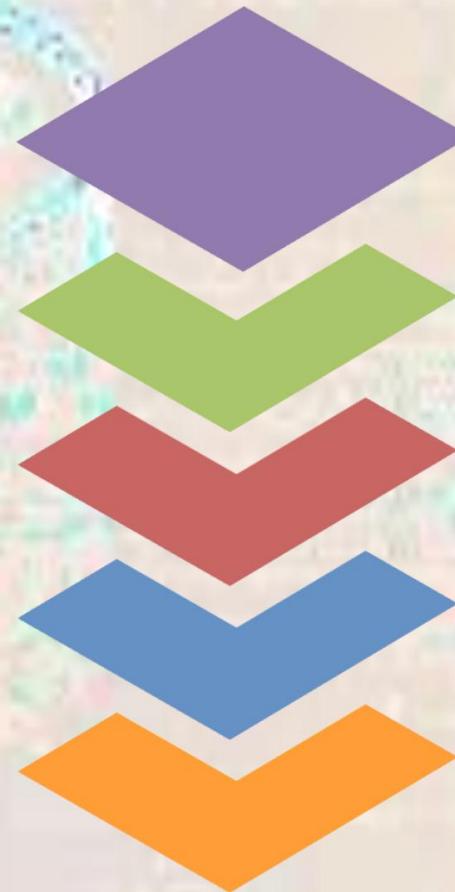
CHỌN ProductID, ProductName,  
Tỷ lệ

TỪ dbo.Product\_Details  
ĐI

# Sử dụng sp\_refreshview

Trong quá trình tạo ra view,  
SCHEMABINDING

tùy chọn được sử dụng để liên kết  
chế độ xem với lược đồ của các  
bảng có trong chế độ xem.



Nếu có thay đổi đối với các đối  
tượng cơ bản (bảng hoặc chế độ xem)  
mà chế độ xem phụ thuộc vào, thì  
quy trình lưu trữ sp\_refreshview sẽ  
được thực thi.

Quy trình lưu trữ sp\_refreshview  
cập nhật siêu dữ liệu cho chế độ xem.

# Thủ tục lưu trữ

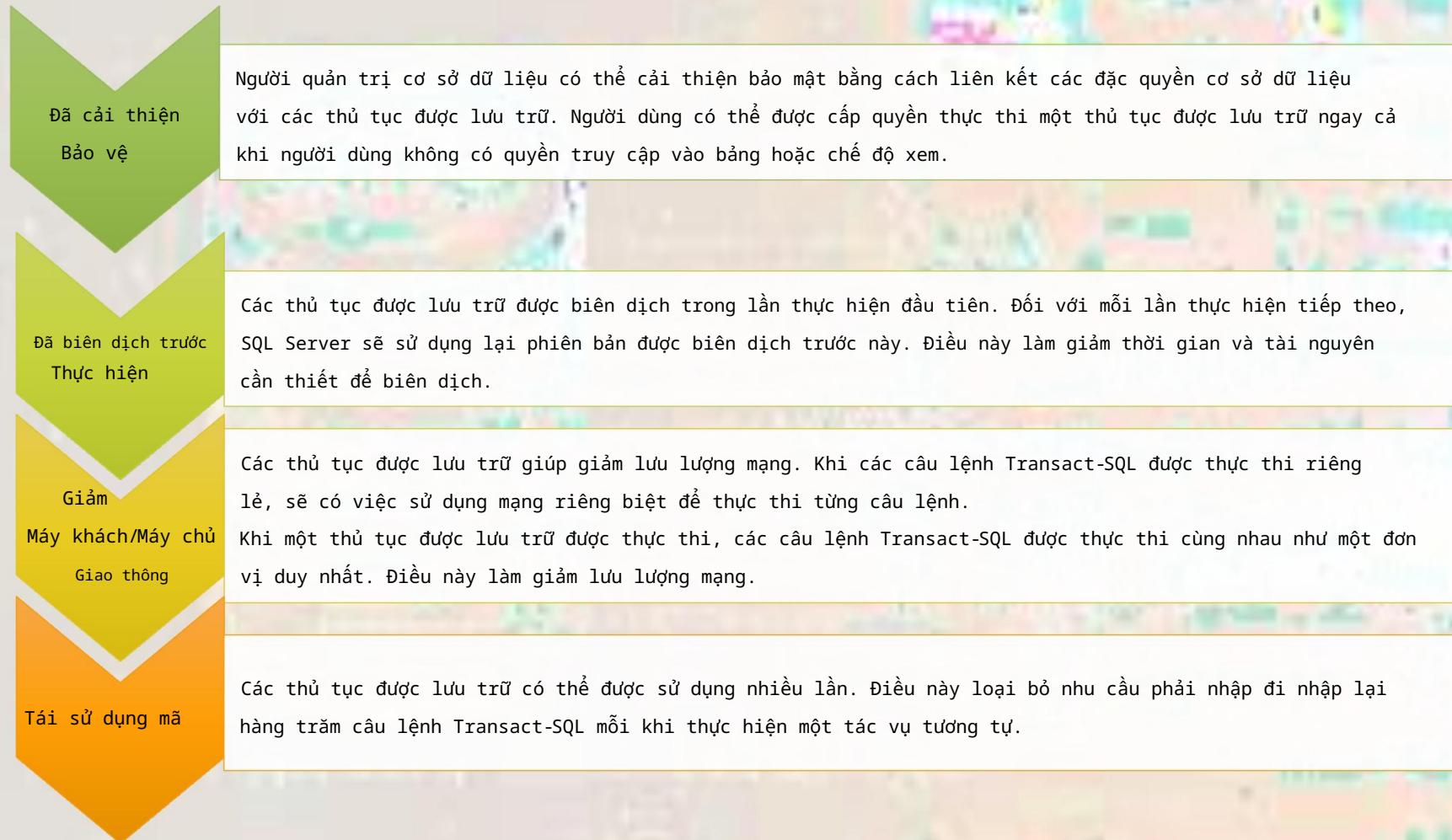
Một thủ tục được lưu trữ là một nhóm các câu lệnh Transact-SQL hoạt động như một khối mã duy nhất thực hiện một tác vụ cụ thể.

Khối mã này được xác định bằng tên được chỉ định và là được lưu trữ trong cơ sở dữ liệu dưới dạng đã biên dịch.

Một thủ tục được lưu trữ cũng có thể là một tham chiếu đến phương thức .NET Framework Common Language Runtime (CLR).

Các thủ tục được lưu trữ hữu ích khi phải thực hiện các tác vụ lặp đi lặp lại. Điều này loại bỏ nhu cầu phải nhập nhiều câu lệnh Transact-SQL rồi biên dịch chúng nhiều lần.

# Thủ tục lưu trữ-2



# Các loại thủ tục lưu trữ 1-3

SQL Server hỗ trợ nhiều loại thủ tục được lưu trữ như

BẰNG:

- **StoredProcedures** do người dùng xác định
- Thủ tục lưu trữ mở rộng

**StoredProcedures** do người dùng định nghĩa

Còn được gọi là thủ tục lưu trữ tùy chỉnh

Được sử dụng để tái sử dụng các câu lệnh Transact-SQL để thực hiện các tác vụ lặp đi lặp lại nhiệm vụ

Hai loại thủ tục lưu trữ do người dùng định nghĩa

Các thủ tục được lưu trữ  
Transact-SQL

Ngôn ngữ chung  
Các thủ tục được lưu trữ thời gian chạy (CLR).

# Các loại thủ tục lưu trữ 2-3

## Thủ tục lưu trữ mở rộng

Giúp SQL Server tương tác với hệ điều hành  
hệ thống

Không phải là đối tượng lưu trú của SQL Server

Chúng là những thủ tục được thực hiện như  
Thư viện liên kết động (DLL) được thực hiện bên ngoài  
Môi trường SQL Server

Sử dụng tiền tố 'xp'

# Các loại thủ tục lưu trữ 3-3

Quy trình lưu trữ hệ thống

Được sử dụng để tương tác với các bảng hệ thống và thực hiện các tác vụ quản trị như cập nhật các bảng hệ thống

Các thủ tục này nằm trong cơ sở dữ liệu Tài nguyên

Các thủ tục được lưu trữ của hệ thống cho phép CẤP, TỪ CHỐI và THU HỒI quyền

Một thủ tục được lưu trữ hệ thống

- Là tập hợp các câu lệnh Transact-SQL được biên dịch trước được thực thi như một đơn vị duy nhất.
- Được sử dụng trong các hoạt động quản lý cơ sở dữ liệu và thông tin.
- Cung cấp quyền truy cập dễ dàng vào thông tin siêu dữ liệu về các đối tượng cơ sở dữ liệu như bảng hệ thống, bảng do người dùng định nghĩa, dạng xem và chỉ mục.

# Phân loại các thủ tục lưu trữ hệ thống

## Danh mục đã lưu trữ Thủ tục

Tất cả thông tin về các bảng trong cơ sở dữ liệu người dùng được lưu trữ trong một tập hợp các bảng được gọi là danh mục hệ thống. Thông tin từ danh mục hệ thống có thể được truy cập bằng các thủ tục danh mục. Ví dụ, thủ tục lưu trữ danh mục sp\_tables hiển thị danh sách tất cả các bảng trong cơ sở dữ liệu hiện tại.

## Bảo mật được lưu trữ Thủ tục

Các thủ tục lưu trữ bảo mật được sử dụng để quản lý bảo mật của cơ sở dữ liệu. Ví dụ, thủ tục lưu trữ bảo mật sp\_changedbowner được sử dụng để thay đổi chủ sở hữu của cơ sở dữ liệu hiện tại.

## Thủ tục lưu trữ con trỏ

Các thủ tục con trỏ được sử dụng để triển khai chức năng của con trỏ. Ví dụ, thủ tục lưu trữ con trỏ sp\_cursor\_list liệt kê tất cả các con trỏ được kết nối mở và mô tả các thuộc tính của chúng.

## Lưu trữ truy vấn phân tán Thủ tục

Các thủ tục lưu trữ phân tán được sử dụng trong việc quản lý các truy vấn phân tán. Đối với Ví dụ, thủ tục lưu trữ truy vấn phân tán sp\_indexes trả về thông tin chỉ mục cho bảng từ xa được chỉ định.

## Cơ sở dữ liệu Mail và SQL Thủ tục lưu trữ thư

Các thủ tục lưu trữ Database Mail và SQL Mail được sử dụng để thực hiện các hoạt động email từ bên trong SQL Server. Ví dụ, thủ tục lưu trữ database mail sp\_send\_dbmail gửi tin nhắn email đến những người nhận được chỉ định. Tin nhắn có thể bao gồm một tập kết quả truy vấn hoặc tệp đính kèm cả hai.

# Thủ tục lưu trữ tạm thời

## Thủ tục tạm thời tại địa phương

## Thủ tục tạm thời toàn cầu

Chỉ hiển thị với người dùng đã tạo ra nó	Hiển thị cho tất cả người dùng
Đã bị loại bỏ vào cuối phiên hiện tại	Đã bỏ vào cuối phiên cuối cùng
Thủ tục tạm thời tại địa phương	Thủ tục tạm thời toàn cầu
Chỉ có thể được sử dụng bởi chủ sở hữu của nó	Có thể được sử dụng bởi bất kỳ người dùng nào
Sử dụng tiền tố # trước tên thủ tục Sử dụng tiền tố ## trước tên thủ tục	

# Quy trình lưu trữ từ xa

Các thủ tục được lưu trữ chạy trên SQL Server từ xa được gọi là thủ tục được lưu trữ từ xa.

Chỉ có thể sử dụng các thủ tục lưu trữ từ xa khi máy chủ từ xa cho phép truy cập từ xa.

Khi một thủ tục lưu trữ từ xa được thực thi từ một phiên bản SQL Server cục bộ đến máy tính khách hàng, có thể gặp phải lỗi hủy bỏ câu lệnh.

Khi lỗi như vậy xảy ra, câu lệnh gây ra lỗi sẽ bị chấm dứt, nhưng quy trình từ xa vẫn tiếp tục được thực thi.

# Thủ tục lưu trữ mở rộng

Các thủ tục lưu trữ mở rộng được sử dụng để thực hiện các tác vụ không thể thực hiện được bằng các câu lệnh Transact-SQL chuẩn.

Các thủ tục lưu trữ mở rộng sử dụng tiền tố 'xp\_'.

Các thủ tục được lưu trữ này được chứa trong lược đồ dbo của cơ sở dữ liệu chính.

# Quy trình lưu trữ tùy chỉnh hoặc do người dùng xác định

Trong SQL Server, người dùng được phép tạo các quy trình lưu trữ tùy chỉnh để thực hiện nhiều tác vụ khác nhau.

Các thủ tục được lưu trữ như vậy được gọi là do người dùng xác định hoặc các thủ tục lưu trữ tùy chỉnh.

	CustomerID	TerritoryID	Name
1	15	9	Australia
2	33	9	Australia
3	51	9	Australia
4	69	9	Australia
5	87	9	Australia
6	105	9	Australia
7	123	9	Australia
8	141	9	Australia
9	159	9	Australia
10	177	9	Australia

Đầu ra của một thủ tục lưu trữ đơn giản

# Sử dụng tham số

Có hai loại tham số:

Đầu vào  
Các tham số

Cho phép chương trình gọi truyền  
giá trị cho một thủ tục  
được lưu trữ. Các giá trị này  
được chấp nhận vào các biến  
được xác định trong thủ  
tục được lưu trữ.

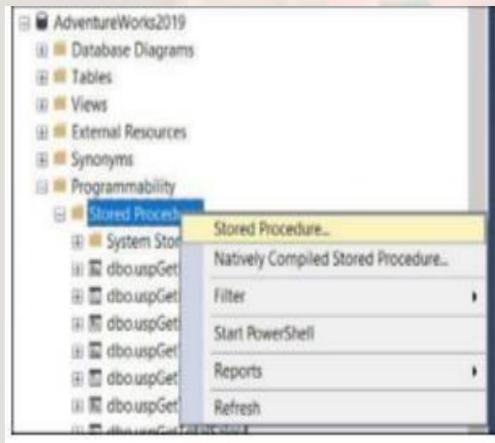
Đầu ra  
Các tham số

Cho phép một thủ tục được lưu trữ  
truyền giá trị trở lại chương  
trình gọi. Các giá trị này  
được chương trình gọi chấp  
nhận vào các biến.

# Sử dụng SSMS để tạo thủ tục lưu trữ

Các bước để tạo thủ tục lưu trữ do người dùng xác định bằng SSMS là:

1. Khởi chạy Object Explorer.
2. Trong Object Explorer, kết nối với một phiên bản của Database Engine và sau khi kết nối thành công, hãy mở rộng phiên bản đó.
3. Mở rộng nút Cơ sở dữ liệu và sau đó mở rộng AdventureWorks2019 cơ sở dữ liệu.



Tạo một StoredProcedure

Parameter	Value
Author	Your name
Create Date	Today's date
Description	Returns year to sales data for a territory
Procedure_Name	uspGetTotals
@Param1	@territory
@Datatype_For_Param1	varchar(50)
Default_Value_For_Param1	NULL
@Param2	
@Datatype_For_Param2	
Default_Value_For_Param2	

Tham số Giá trị

# Sửa đổi và xóa các thủ tục lưu trữ

Quyền liên quan đến thủ tục lưu trữ bị mất khi thủ tục lưu trữ được tạo lại. Tuy nhiên, khi thủ tục lưu trữ bị thay đổi, quyền được xác định cho thủ tục lưu trữ vẫn giữ nguyên ngay cả khi định nghĩa thủ tục bị thay đổi.

## Xóa bỏ các thủ tục lưu trữ

Có thể xóa các thủ tục lưu trữ nếu chúng không còn cần thiết nữa. Nếu một thủ tục lưu trữ khác gọi một thủ tục đã xóa, một thông báo lỗi sẽ được hiển thị.

Nếu một thủ tục mới được tạo bằng cùng tên cũng như các tham số giống như thủ tục đã bị loại bỏ, tất cả các lệnh gọi đến thủ tục đã bị loại bỏ sẽ được thực hiện thành công.

# Thủ tục lưu trữ lồng nhau

SQL Server 2019 cho phép gọi các thủ tục được lưu trữ bên trong các thủ tục được lưu trữ khác.

Các thủ tục được gọi có thể lần lượt gọi các thủ tục khác.

Kiến trúc gọi một thủ tục từ một thủ tục khác được gọi là kiến trúc thủ tục lưu trữ lồng nhau.

Khi một thủ tục được lưu trữ gọi một thủ tục được lưu trữ khác, mức độ lồng nhau được cho là tăng thêm một.

Tương tự như vậy, khi một thủ tục được gọi hoàn tất quá trình thực thi và chuyển quyền điều khiển trở lại cho thủ tục gọi, mức độ lồng nhau được cho là giảm đi một.

Mức lồng nhau tối đa được SQL Server 2019 hỗ trợ là 32.

# Hệ thống truy vấn MetaData1-3

Các thuộc tính của một đối tượng như bảng hoặc dạng xem được lưu trữ trong các bảng hệ thống đặc biệt.

Các thuộc tính này được gọi là siêu dữ liệu. Tất cả các đối tượng SQL đều tạo ra siêu dữ liệu.

Siêu dữ liệu này có thể được xem bằng cách sử dụng chế độ xem hệ thống, là chế độ xem được xác định trước của SQLServer

## Danh mục hệ thốngXem

Chúng chứa thông tin về danh mục trong hệ thống SQL Server.

Danh mục tương tự như một bản kiểm kê các đồ vật.

Các chế độ xem này chứa nhiều siêu dữ liệu khác nhau.

# Hệ thống truy vấn MetaData2-3

## Sơ đồ thông tin

Người dùng có thể truy vấn chế độ xem lược đồ thông tin để trả về siêu dữ liệu hệ thống.

Các chế độ xem này hữu ích cho các công cụ của bên thứ ba có thể không dành riêng cho SQL Server.

Chế độ xem lược đồ thông tin cung cấp chế độ xem nội bộ, độc lập với bảng hệ thống về siêu dữ liệu SQL Server.

Lượt xem sơ đồ thông tin	Chế độ xem hệ thống SQL Server
Chúng được lưu trữ trong INFORMATION_SCHEMA. lược đồ riêng của họ,	Chúng xuất hiện trong lược đồ hệ thống .
Họ sử dụng thuật ngữ chuẩn thay vì SQL Server thuật ngữ. Ví dụ, họ sử dụng danh mục thay vì cơ sở dữ liệu. Họ tuân và miên thay vì kiểu dữ liệu do người dùng định nghĩa.	thủ thuật ngữ SQL Server.
Họ có thể không hiển thị tất cả siêu dữ liệu có sẵn cho chế độ xem danh mục riêng của SQL Server. Ví dụ, sys.columns bao gồm các thuộc tính cho thuộc tính nhận dạng và tính toán thuộc tính cột, trong khi INFORMATION_SCHEMA.columns thì không.	Họ có thể hiển thị tất cả siêu dữ liệu có sẵn cho chế độ xem danh mục của SQL Server.

Thông tin Schema Views and SQL Server System Views

# Hệ thống truy vấn MetaData3-3

## Siêu dữ liệu hệ thốngChức năng

Các hàm tích hợp có thể trả về siêu dữ liệu cho truy vấn.  
 Chúng bao gồm các hàm vô hướng và các hàm có giá trị bảng. Các hàm siêu dữ liệu của SQL Server có nhiều định dạng khác nhau.

Tên hàm	Mô tả	Ví dụ
OBJECT_ID(<tên đối tượng>)	đối tượng của OBJECT_ID('Sales.Custom database object.er')	
TÊN_ĐỐI_TƯỢNG(<id đối tượng>)	Trả về tên OBJECT_ tương ứng với ID đối tượng.	TÊN(197575742)
@@LỖI	Trả về 0 nếu câu lệnh cuối cùng thành công; nếu không trả về số lỗi.	@@LỖI
THUỘC TÍNH MÁY CHỦ(< thuộc tính>)	Trả về giá trị của thuộc tính máy chủ được chỉ định.	THUỘC TÍNH MÁY CHỦ('Đối chiếu')

Các chức năng siêu dữ liệu hệ thống chung

# Truy vấn đối tượng DynamicManagement

Chế độ xem quản lý động (DMV) và Chức năng quản lý động (DMF) là các đối tượng quản lý động trả về thông tin trạng thái của máy chủ và cơ sở dữ liệu.

DMV và DMF được gọi chung là đối tượng quản lý động.

Được sử dụng để kiểm tra trạng thái của phiên bản SQL Server, khắc phục sự cố và điều chỉnh hiệu suất.

Để truy vấn DMV, cần phải có quyền VIEW SERVER STATE hoặc VIEW DATABASE STATE, tùy thuộc vào phạm vi của DMV.

# Phân loại và truy vấn DMV

Đặt tên mẫu db	Sự miêu tả
	Cơ sở dữ liệu liên quan
tối	Thống kê I/O
Hệ điều hành	Hệ điều hành SQL Server Thông tin
'tran'	Liên quan đến giao dịch
'exec'	Siêu dữ liệu liên quan đến thực hiện truy vấn

Tổ chức DMVs theo Chức năng

	Results	Messages
1	session_id 54	login_time 2020-11-11 08:27:27.390
2	55	2020-11-11 08:27:37.180
3	56	2020-11-11 08:27:42.287

Truy vấn sys.dm\_exec\_sessionsDMV

# Bản tóm tắt

- Một chế độ xem là một bảng ảo được tạo thành từ các cột được chọn từ một hoặc nhiều bảng và được tạo bằng lệnh CREATE VIEW trong SQL Server. • Người dùng có thể thao tác dữ liệu trong chế độ xem, chẳng hạn như chèn vào chế độ xem, sửa đổi dữ liệu trong chế độ xem và xóa khỏi chế độ xem. • Một thủ tục được lưu trữ là một nhóm các câu lệnh Transact-SQL hoạt động như một khôi phục thực hiện một nhiệm vụ cụ thể.
- SQL Server hỗ trợ nhiều loại thủ tục được lưu trữ khác nhau, chẳng hạn như Thủ tục được lưu trữ do người dùng xác định, Thủ tục được lưu trữ mở rộng và Thủ tục được lưu trữ hệ thống. • Thủ tục được lưu trữ hệ thống có thể được phân loại thành các loại khác nhau như Thủ tục được lưu trữ danh mục, Thủ tục được lưu trữ bảo mật và Thủ tục được lưu trữ con trỏ.
- Các tham số đầu vào và đầu ra có thể được sử dụng với các thủ tục được lưu trữ để truyền và nhận dữ liệu từ các thủ tục được lưu trữ. • Các thuộc tính của một đối tượng như bảng hoặc chế độ xem được lưu trữ trong các bảng hệ thống đặc biệt và được tham chiếu đến dưới dạng siêu dữ liệu. • DMV và DMF là các đối tượng quản lý động trả về thông tin trạng thái máy chủ và cơ sở dữ liệu. DMV và DMF được gọi chung là các đối tượng quản lý động.