

VIEW – KHUNG NHÌN

Contents



Giới thiệu view

Tạo View

Sửa View

Xóa View

Partitioned Views

Định nghĩa

- Một khung nhìn (view) có thể được xem như là một bảng “ảo” trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một truy vấn (câu lệnh SELECT).
- Một khung nhìn là một tập bao gồm các dòng và các cột.
- Khung nhìn không được xem là một cấu trúc lưu trữ dữ liệu tồn tại trong cơ sở dữ liệu.
- Dữ liệu quan sát được trong khung nhìn được lấy từ các bảng thông qua câu lệnh truy vấn dữ liệu và là kết quả động khi view được tham chiếu.

Thuận lợi khi sử dụng view

- ❖ Bảo mật dữ liệu: **Chỉ cho User xem những gì cần xem nên hạn chế được phần nào việc người sử dụng truy cập trực tiếp dữ liệu.**
- ❖ Đơn giản hoá các thao tác truy vấn dữ liệu: **Một khung nhìn là một đối tượng tập hợp dữ liệu từ nhiều bảng khác nhau vào trong một “bảng”. User có thể thực hiện các yêu cầu truy vấn dữ liệu một cách đơn giản thay vì phải dùng truy vấn phức tạp.**
- ❖ Tập trung và đơn giản hóa dữ liệu: **cung cấp cho người sử dụng những cấu trúc đơn giản, dễ hiểu hơn về dữ liệu trong CSDL đồng thời giúp cho người sử dụng tập trung hơn trên những phần dữ liệu cần thiết.**
- ❖ Độc lập dữ liệu: **người sử dụng có được cái nhìn về dữ liệu độc lập với cấu trúc của các bảng trong CSDL cho dù các bảng cơ sở có bị thay đổi phần nào về cấu trúc.**
- ❖ Dùng để Import, Export

Thuận lợi khi sử dụng view

MAHD

1

NGÀY BÁN

MAKH

A

TEN KH

LAN

TEN NV

MINH

STT	MAHH	TENHH	SL	DON GIÁ	THÀNH TIỀN
1	P1	TÁO	10	100	1000
2	P2	CAM	5	50	250

Thuận lợi khi sử dụng view

MASV	HODEM	TEN	NGAYSINH
0241010001	Ngô Thị Nhật	Anh	Nov 27 1982 ...
0241010002	Nguyễn Thị Ngọc	Anh	Mar 21 1983 ...
0241010003	Ngô Việt	Bắc	May 11 1982 ...
0241010004	Nguyễn Đình	Bình	Oct 6 1982 ...
0241010005	Hồ Đăng	Chiến	Jan 20 1982 ...
0241020001	Nguyễn Tuấn	Anh	Jul 15 1979 ...
0241020002	Trần Thị Kim	Anh	Nov 4 1982 ...
...

Table SINHVIEN

MALOP	TENLOP	
C24101	Toán K24	...
C24102	Tin K24	...
C24103	Lý K24	...
..

Table LOP



MASV	HODEM	TEN	TUOI	TENLOP
0241010001	Ngô Thị Nhật	Anh	22	Toán K24
0241010002	Nguyễn Thị Ngọc	Anh	21	Toán K24
0241010003	Ngô Việt	Bắc	22	Toán K24
0241010004	Nguyễn Đình	Bình	22	Toán K24
0241010005	Hồ Đăng	Chiến	22	Toán K24
0241020001	Nguyễn Tuấn	Anh	25	Tin K24
0241020002	Trần Thị Kim	Anh	22	Tin K24
...

view DSSV

Hạn chế khi sử dụng View

- Không bao gồm các mệnh đề COMPUTER hoặc COMPUTER BY.
- Không bao gồm từ khóa INTO.
- Chỉ được dùng ORDER BY khi từ khóa TOP được dùng.
- Không thể tham chiếu quá 1024 cột.
- Không thể kết hợp với câu lệnh T-SQL khác trong cùng một bó lệnh.
- Không thể định nghĩa chỉ mục full text trên View.

Tạo View

Cú pháp

```
CREATE VIEW [<db_name>.] [<owner>.] view_name [(column[ ,...n ])]  
[WITH <view_attribute>[,...n]]  
AS <Select_Statement>  
[WITH CHECK OPTION]  
<view_attribute>::=  
    { ENCRYPTION | SCHEMABINDING }
```

WITH CHECK OPTION: bắt buộc tất cả các lệnh hiệu chỉnh dữ liệu của View phải thỏa mãn các tiêu chuẩn trong câu lệnh Select.

ENCRYPTION: Mã hóa câu lệnh Select tạo ra View.

SCHEMABINDING: Kết View với giản đồ

Tạo View

❖ Ví dụ:

CREATE VIEW vwProducts

AS

**SELECT ProductName, UnitPrice,
CompanyName**

FROM Suppliers

INNER JOIN Products

**ON Suppliers.SupplierID =
Products.SupplierID**

Tạo View

Ví dụ

```
CREATE VIEW CTHD AS  
SELECT Orderid, Products.Productid, Productname,  
       Quantity, UnitPrice, ToTal = UnitPrice *Quantity  
FROM Products INNER JOIN [Order Details]  
ON Products.Productid = [Order Details].Productid
```

Nguyên tắc tạo View

- ❖ Tên khung nhìn, tên cột trong View và bảng phải tuân theo qui tắc định danh.
- ❖ Không thể qui định ràng buộc và tạo chỉ mục cho khung nhìn.
- ❖ Câu lệnh SELECT với mệnh đề COMPUTE ... BY không được sử dụng để định nghĩa khung nhìn.
- ❖ Phải đặt tên cho các cột của khung nhìn trong các trường hợp sau:
 - Trong kết quả của câu lệnh SELECT có ít nhất một cột được sinh ra bởi một biểu thức và cột đó không được đặt tiêu đề.
 - Tồn tại hai cột trong kết quả của câu lệnh SELECT có cùng tiêu đề cột.

Nguyên tắc tạo View

❖ Ví dụ 1:

```
CREATE VIEW dsnv AS
```

```
SELECT Employees.EmployeeID,FirstName+'  
      '+LastName AS HOTEN,  
DATEDIFF(YY,birthdate,GETDATE()) AS tuoi  
FROM Employees
```

	EmployeeID	HOTEN	tui
1	1	Nancy Davolio	68
2	2	Andrew Fuller	64
3	3	Janet Leverling	53
4	4	Margaret Peacock	79
5	5	Steven Buchanan	61
6	6	Michael Suyama	53
7	7	Robert King	56
8	8	Laura Callahan	58
9	9	Anne Dodsworth	50

Nguyên tắc tạo View

❖ Ví dụ 2:

```
CREATE VIEW dsnv (MANV, HOTEN, TUOI) AS  
SELECT Employees.EmployeeID, FirstName+'  
        '+LastName AS HOTEN,  
DATEDIFF(YY,birthdate,GETDATE()) AS tuoi  
FROM Employees
```

	MANV	HOTEN	TUOI
1	1	Nancy Davolio	68
2	2	Andrew Fuller	64
3	3	Janet Leverling	53
4	4	Margaret Peacock	79
5	5	Steven Buchanan	61
6	6	Michael Suyama	53
7	7	Robert King	56
8	8	Laura Callahan	58
9	9	Anne Dodsworth	50

Nguyên tắc tạo View

❖ Ví dụ 3:

```
CREATE VIEW TuoivNv AS
```

```
SELECT Employees.EmployeeID,FirstName+'  
      '+LastName AS HOTEN,  
DATEDIFF(YY,birthdate,GETDATE()) AS tuoi  
FROM Employees
```

	EmployeeID	HOTEN	tui
1	1	Nancy Davolio	68
2	2	Andrew Fuller	64
3	3	Janet Leverling	53
4	4	Margaret Peacock	79
5	5	Steven Buchanan	61
6	6	Michael Suyama	53
7	7	Robert King	56
8	8	Laura Callahan	58
9	9	Anne Dodsworth	50

Tạo View với *ENCRYPTION*

- ❖ With *ENCRYPTION* : Mã hóa câu lệnh Select tạo ra View.

```
CREATE VIEW vwProducts  
WITH ENCRYPTION  
AS  
SELECT CompanyName, ProductName, UnitPrice  
FROM Suppliers INNER JOIN Products  
ON Suppliers.SupplierID = Products.SupplierID  
GO  
EXEC sp_helptext vwProducts
```

Tạo View với *SCHEMABINDING*

- ❖ With *SCHEMABINDING*: Kết view với một giản đồ. Khi *SCHEMABINDING* được chỉ định, câu lệnh Select phải chỉ rõ chủ quyền của các bảng, các view. Các hàm được tham chiếu View hay bảng tham gia trong view được tạo với schema không thể xóa trừ phi View đó bị xóa hay thay đổi cơ chế này. Câu lệnh Alter table trên bảng tham gia trong view cũng bị lỗi.

```
CREATE VIEW vwProducts  
WITH SCHEMABINDING  
AS
```

```
    SELECT CompanyName, ProductName, UnitPrice  
    FROM dbo.Suppliers INNER JOIN dbo.Products  
    ON Suppliers.SupplierID = Products.SupplierID
```

```
GO
```

```
ALTER TABLE dbo.Products  
DROP COLUMN UnitPrice
```


Tạo View với lựa chọn Check

Bắt buộc tất cả các câu lệnh hiệu chỉnh dữ liệu thực thi dựa vào View phải tuyệt đối tôn trọng triệt để đến tập tiêu chuẩn trong câu lệnh Select. Nếu không dùng CHECK, các dòng không thể được hiệu chỉnh. Bất kỳ hiệu chỉnh nào mà sẽ gây ra tình trạng thay đổi đều bị hủy bỏ và một lỗi được hiện ra.

- ❖ CREATE VIEW CustomersCAView AS

 - SELECT * FROM Customers WHERE city='LonDon'

- ❖ Select * from CustomersCAView

- ❖ GO

- ❖ UPDATE CustomersCAView SET city='Anh Quoc'
WHERE CustomerID='AROUT'

- ❖ select * from Customers where CustomerID='AROUT'

Tạo View với lựa chọn Check

```
CREATE VIEW CustomersCAView1  
AS  
SELECT * FROM Customers WHERE city='LonDon'  
WITH CHECK OPTION
```

```
Select * from CustomersCAView1  
GO
```

```
UPDATE CustomersCAView1 SET city='Anh Quoc'  
WHERE CustomerID='NORTS'
```

Cập nhật, bổ sung và xoá dữ liệu thông qua View

- ❖ Các thao tác bổ sung, cập nhật và xoá, một khung nhìn phải thoả mãn các điều kiện sau đây:
 - Trong câu lệnh SELECT định nghĩa khung nhìn không được sử dụng từ khoá DISTINCT, TOP, GROUP BY và UNION.
 - Các thành phần xuất hiện trong danh sách chọn của câu lệnh SELECT phải là các cột trong các bảng cơ sở. Trong danh sách chọn không được chứa các biểu thức tính toán, các hàm gộp.
- ❖ Các thao tác thay đổi đến dữ liệu thông qua khung nhìn còn phải đảm bảo tính toàn vẹn dữ liệu.

Cập nhật dữ liệu thông qua View

- ❖ Ví dụ 3.14: Xét định nghĩa hai bảng DONVI và NHANVIEN như sau:

```
CREATE TABLE donvi
(  madv INT PRIMARY KEY,
   tendv NVARCHAR(30) NOT NULL,
   dienthoai NVARCHAR(10) NULL
)
CREATE TABLE nhanvien
(  manv NVARCHAR(10) PRIMARY KEY,
   hoten NVARCHAR(30) NOT NULL,
   ngaysinh DATETIME NULL,
   diachi NVARCHAR(50) NULL,
   madv INT FOREIGN KEY
   REFERENCES donvi(madv)
   ON DELETE CASCADE
   ON UPDATE CASCADE
)
```

Cập nhật dữ liệu thông qua View

❖ Ví dụ: Xét định nghĩa hai bảng DONVI và NHANVIEN như sau:
Insert into DonVi (Madv, Tendv, DiuenThoai) values (1,'P.Kinh doanh','822321')

Insert into DonVi (Madv, Tendv, DiuenThoai) values (2,Tiep thi','822012')

Insert into nhanvien(manv,hoten,ngaysinh,diachi,madv)
Values('NV01','Tran Van A','3/2/1975','77 Tran Phu',1)

Insert into nhanvien(manv,hoten,ngaysinh,diachi,madv)
Values('NV02','Mai Thi Bich','13/2/1977','17 Nguyen Hue',2)

Insert into nhanvien(manv,hoten,ngaysinh,diachi,madv)
Values('NV03','Le Van Ha','3/2/1973','12 Tran Phu',2)

MADV	TENDV	DIENTHOAI
1	P. Kinh doanh	822321
2	P. Tiep thi	822012

Bảng DONVI

MANV	HOTEN	NGAYSINH	DIACHI	MADV
NV01	Tran Van A	1975-02-03 00:00:00	77 Tran Phu	1
NV02	Mai Thi B	1977-05-04 00:00:00	34 Nguyen Hue	2
NV03	Nguyen Van C	NULL	NULL	2

Bảng NHANVIEN

Cập nhật, bổ sung và xoá dữ liệu thông qua View

```
CREATE VIEW nv1
```

```
AS
```

```
    SELECT manv,hoten,madv FROM nhanvien
```

```
GO
```

```
INSERT INTO nv1 VALUES('NV04','Le Thi D',1)
```

MANV	HOTEN	NGAYSINH	DIACHI	MADV
NV01	Tran Van A	1975-02-03 00:00:00	77 Tran Phu	1
NV02	Mai Thi B	1977-05-04 00:00:00	34 Nguyen Hue	2
NV03	Nguyen Van C	NULL	NULL	2
NV04	Le Thi D	NULL	NULL	1

Bản ghi mới



```
DELETE FROM nv1 WHERE manv='NV04'
```

Cập nhật dữ liệu thông qua View

- ❖ Nếu câu lệnh **SELECT** có sự xuất hiện của biểu thức tính toán đơn giản, thao tác bổ sung dữ liệu thông qua khung nhìn có thể thực hiện được. Thao tác cập nhật và xóa dữ liệu vẫn có thể có khả năng thực hiện được (trừ cột là một biểu thức tính toán).

- ❖ Ví dụ : Xét khung nhìn NV2 được định nghĩa như sau:

```
CREATE VIEW nv2
```

```
AS
```

```
    SELECT manv,hoten,YEAR(ngaysinh) AS namsinh,madv FROM nhanvien
```

```
GO
```

```
INSERT INTO nv2(manv,hoten,madv) VALUES('NV05','Le Van E',1) –Lỗi
```

```
GO
```

```
UPDATE nv2 SET hoten='Le Thi X' WHERE manv='NV04' –Thực hiện được
```

```
GO
```

```
DELETE FROM nv2 WHERE manv='NV04' –Thực hiện được
```

Cập nhật dữ liệu thông qua View

- ❖ Nếu khung nhìn được tạo ra từ một phép nối (trong hoặc ngoài) trên nhiều bảng, ta có thể thực hiện được thao tác bổ sung hoặc cập nhật dữ liệu nếu thao tác này chỉ có tác động đến đúng một bảng cơ sở (câu lệnh DELETE không thể thực hiện được trong trường hợp này).
- ❖ Ví dụ: Với khung nhìn được định nghĩa như sau:

```
CREATE VIEW nv3
```

```
AS
```

```
SELECT  manv,hoten,ngaysinh,  diachi,nhanvien.madv  AS  noilamviec,  
        donvi.madv,tendv,dienthoai FROM nhanvien FULL OUTER JOIN donvi  
        ON nhanvien.madv=donvi.madv
```

```
GO
```

```
--Thêm vào bảng NHANVIEN
```

```
INSERT INTO nv3(manv,hoten,noilamviec) VALUES('NV05','Le Van E',1)
```

```
--Thêm vào bảng DONVI
```

```
INSERT INTO nv3(madv,tendv) VALUES(3,'P. Ke toan')
```


Bổ sung dữ liệu thông qua View

❖ Cú pháp:

ALTER VIEW *tên_khung_nhìn* [(*danh_sách_tên_cột*)]
AS

Câu_lệnh_SELECT

❖ Ví dụ: Ta định nghĩa khung nhìn như sau:

❖ CREATE VIEW viewDV

❖ AS

❖ SELECT manv,hoten,tendv

❖ FROM donvi INNER JOIN nhanvien ON donvi.madv=nhanvien.madv

❖ WHERE tendv='P.Kinh doanh'

❖

❖ select * from viewDV

❖ drop view viewDV

❖ ALTER VIEW viewDV

❖ AS

❖ SELECT manv,hoten,tendv

❖ FROM donvi INNER JOIN nhanvien ON donvi.madv=nhanvien.madv

❖ WHERE tendv='Tiep Thi'

Xóa View

❖ Cú pháp:

`DROP VIEW tên_khung_nhìn`

❖ Nếu một khung nhìn bị xóa, toàn bộ những quyền đã cấp phát cho người sử dụng trên khung nhìn cũng đồng thời bị xóa. Do đó, nếu ta tạo lại khung nhìn thì phải tiến hành cấp phát lại quyền cho người sử dụng.

❖ Ví dụ: `DROP VIEW viewDV`

Đổi tên Views

❖ **Đổi tên Views:**

Cú pháp:

`sp_rename old_viewname, new_viewname`

Ví dụ : `Sp_rename CTHD, ChiTietHD`

❖ **Xác nhận Views:**

Cú pháp:

`sp_helptext viewname`

Ví dụ : `Sp_helptext ChitietHD`

Các loại Views

- ❖ **Standard View**
- ❖ **Indexed View**
- ❖ **Partitioned View**

Các loại Views

❖ Standard View

- Kết hợp dữ liệu từ nhiều bảng tùy theo mục đích sử dụng.
- Các lợi ích khi sử dụng view:
 - ✓ Cung cấp dữ liệu thích hợp cho người dùng
 - ✓ Che giấu sự phức tạp của dữ liệu
 - ✓ Kết hợp dữ liệu từ các nguồn không đồng nhất
 - ✓ View được dùng như 1 cơ chế bảo mật (security mechanism) bằng cách cho phép người dùng truy xuất dữ liệu thông qua view mà không cấp cho người dùng quyền được truy xuất trực tiếp dữ liệu từ bảng gốc.

Các loại Views

❖ Standard View

EmployeeMaster table

EmployeeID	FirstName	AddressID	ShiftID	LastName	MiddleName	SSN	...
1	Sheri	1	1	Nowmer	E	245797967	...
2	Derrick	2	1	Whelply	R	509647174	...
3	Michael	3	1	Spence	C	42487730	...
4	Maya	4	1	Gutierrez	Y	56920285	...
5	Roberta	5	1	Damstra	B	695256908	...

View

FirstName	LastName	Description
Sheri	Nowmer	Engineering
Derrick	Whelply	Engineering
Michael	Spence	Engineering

Department Table

DepartmentID	Description	rowguid
1	Engineering	3FFD2603-EB6E-43B2-A8EF-C4F5C3064026
2	Tool Design	AE948718-D4BF-40E0-8ECD-2D9F4A0B211E
3	Sales	702C0EE3-03E6-4F95-9AB8-99F4F25921F3
4	Marketing	3E3C4476-B9EC-43CB-AA12-1E7A140A71A4
5	Purchasing	D6C63691-93B5-4F43-AD88-34B6B9A3C4A3

Các loại Views

❖ Indexed View

- Là view đã được hiện thực hóa (materialized), nghĩa là view đã được tính toán thực thi và lưu trữ như 1 bảng thực. View có 1 chỉ mục clustered duy nhất.
- Lý do sử dụng indexed view:

Nếu 1 view tham chiếu đến 1 truy vấn phức tạp thì chi phí để tạo dựng bộ kết quả cho truy vấn khi sử dụng view rất lớn. Để cải thiện việc thực thi, cần tạo chỉ mục cho view.
- View chỉ mục không thích hợp cho những bảng dữ liệu hay cập nhật chỉnh sửa

Các loại Views

❖ Indexed View

- Để tạo clustered index cho 1 view, cần phải đáp ứng các yêu cầu sau:
 - Không được tham chiếu đến các view khác, chỉ từ bảng gốc. Tất cả bảng gốc này phải cùng 1 database và có cùng 1 owner.
 - View phải được tạo ra với tùy chọn SCHEMABINDING.
 - Các hàm người dùng được tham chiếu đến trong view phải được ra với tùy chọn SCHEMABINDING.
 - Bảng và các hàm người dùng phải được tham chiếu bởi tên gồm 2 thành phần.

Các loại Views

❖ Indexed View

```
CREATE VIEW Sales.vOrders
WITH SCHEMABINDING
AS
    SELECT SUM(UnitPrice*OrderQty*(1.00-UnitPriceDiscount)) AS Revenue,
           OrderDate, ProductID, COUNT_BIG(*) AS COUNT
    FROM Sales.SalesOrderDetail AS od, Sales.SalesOrderHeader AS o
    WHERE od.SalesOrderID = o.SalesOrderID
    GROUP BY OrderDate, ProductID;
```

```
--Create an index on the view.
CREATE UNIQUE CLUSTERED INDEX IDX_V1
    ON Sales.vOrders (OrderDate, ProductID);
```

Các loại Views

❖ Indexed View

Create View HDKH
WITH SCHEMABINDING
AS

```
Select orderdate,COUNT(*) As ToTal  
From [Customers] c , Orders o  
Where c.CustomerID = o.CustomerID  
Group by OrderDate
```

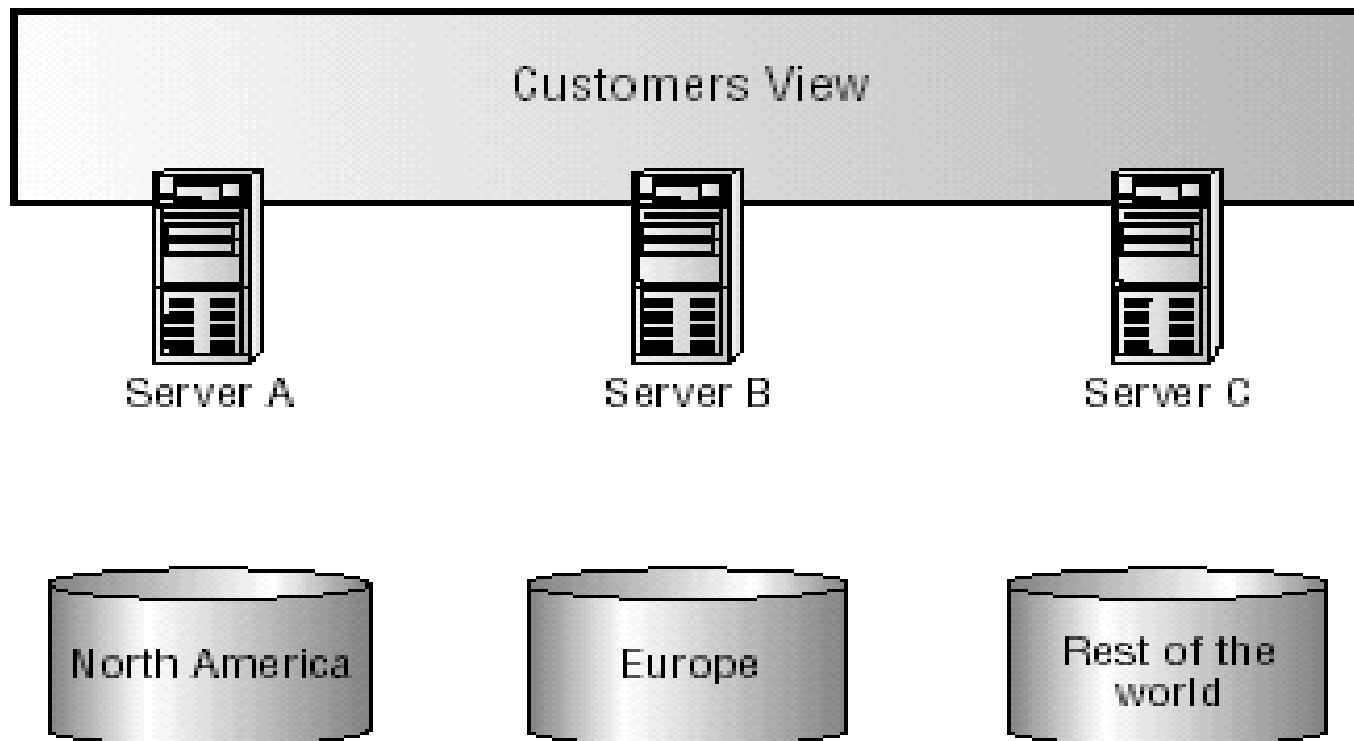
Create UNIQUE CLUSTERED INDEX IDX_V1 ON
SalesOrder(orderdate,Productid);

Các loại Views

❖ Indexed View

```
--This query can use the indexed view even though the view is  
--not specified in the FROM clause.  
SELECT SUM(UnitPrice*OrderQty*(1.00-UnitPriceDiscount)) AS Rev,  
       OrderDate, ProductID  
FROM Sales.SalesOrderDetail AS od  
     JOIN Sales.SalesOrderHeader AS o ON od.SalesOrderID=o.SalesOrderID  
     AND ProductID BETWEEN 700 and 800  
     AND OrderDate >= CONVERT(datetime,'05/01/2002',101)  
GROUP BY OrderDate, ProductID  
ORDER BY Rev DESC;
```

Partitioned Views



Partitioned Views

- Các bảng tham gia Partition view phải có cấu trúc giống nhau.
- Có một cột có check constraint với phạm vi của ràng buộc CHECK ở mỗi bảng là khác nhau.
- Tạo View bằng cách kết các dữ liệu bằng từ khóa UNION ALL.
- Cột là NOT NULL.
- Cột là một phần khóa chính của table.
- Không có cột tính toán.
- Chỉ có duy nhất một ràng buộc CHECK tồn tại trong một cột.
- Bảng không thể có chỉ mục trong các cột tính toán.

Partitioned Views

Ví dụ:

```
CREATE VIEW Customers
```

```
AS
```

```
SELECT * FROM
```

```
ServerA.MyCompany.dbo.CustomersAmerica
```

```
UNION ALL
```

```
SELECT * FROM
```

```
ServerB.MyCompany.dbo.CustomersEurope
```

```
UNION ALL
```

```
SELECT * FROM
```

```
ServerC.MyCompany.dbo.CustomersAsia
```

Partitioned Views

Ví dụ

Create Table KH_BAC

(Makh int, TenKh Nchar(30),

Khuvuc Nvarchar(30) NOT NULL CHECK (Khuvuc='Bac
bo'),

PRIMARY KEY (Makh, Khuvuc)

)

Create Table KH_TRUNG

(Makh int, TenKh Nchar(30),

Khuvuc Nvarchar(30) NOT NULLCHECK (Khuvuc='Trung
bo'),

PRIMARY KEY (Makh, Khuvuc))

Partitioned Views

Create Table KH_NAM

(Makh int, TenKh Nchar(30),

**Khuvuc Nvarchar(30) NOT NULL CHECK
(Khuvuc='Nam bo'),**

PRIMARY KEY (Makh, Khuvuc)

)

Partitioned Views

Create View Khachhang
AS

```
        Select * From KH_BAC  
UNION ALL  
        Select * From KH_TRUNG  
UNION ALL  
        Select * From KH_NAM
```

INSERT Khachhang VALUES (1, 'CDCN4','Nam Bo')

SELECT * FROM KH_Nam

Hiệu chỉnh dữ liệu thông qua Partitioned Views

- Tất cả các cột phải có giá trị ngay cả cột chấp nhận Null và cột có giá trị Default.
- Từ khóa Default không được sử dụng trong câu lệnh Insert, Update.
- Phải có giá trị đúng của cột có ràng buộc CHECK.
- Câu lệnh INSERT không cho phép nếu bảng thành viên có cột có thuộc tính Identity, cột timestamp.
- Không Insert, Update hay Delete nếu có một kết self-join trong cùng View hay bảng thành viên.
- Khi dùng lệnh Delete ta có thể xóa các mẫu tin trong bảng thành viên thông qua View.