

Views

نظرة عامة:

رأينا عند دراسة وتحليل البيانات، أن الاستعلام تقنية تسمح بعزل سلسلة من الحقول و/أو السجلات من الجداول، يتم ذلك عادة لغرض تحليل البيانات. ويمكن أيضا القيام بذلك لأجل إنشاء قائمة جديدة من البيانات لأغراض معينة.

يتم إنشاء الاستعلام غالبا بشكل مؤقت، مثلا خلال تحليل البيانات، باستخدام الجداول، النماذج، صفحات على شبكة الإنترنت. يتم التخلص من الاستعلام مباشرة بعد الحصول على القائمة المطلوبة.

تسمح العديد من تطبيقات قواعد البيانات، بما في ذلك SQL Server، بإنشاء استعلامات مع إمكانية تخزينها لاستخدامها لاحقا، أو حتى لاستخدامها كما لو كانت جداول من قاعدة البيانات. هذه هي الفكرة الأساسية حول طريقة عمل كائنات تدعى Views.

View هي لائحة من الحقول أو مجموعة من السجلات التي تم استردادها من جدول أو أكثر، أو مزيج من View واحدة أو أكثر، أو Views وجداول أخرى. بناء على هذا، قبل إنشاء View يجب أولا تحديد الحقول والسجلات التي سيتم التعامل معها.

تطبيق عملي: التمهيد لإنشاء View

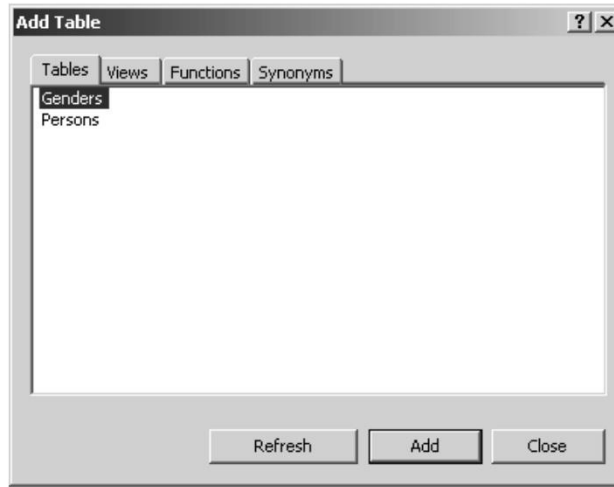
1- قم بتشغيل SQL Server Management Studio واتصل بالخادم

2- تأكد من وجود قاعدة بيانات RealEstate2 التي تم إنشاؤها في درس سابق.

أساسيات قبل إنشاء View:

من أجل إنشاء View، يمكنك استخدام متصفح الكائنات أو إطار الاستعلام. وقبل بدء إنشاء View ينبغي تحديد الجدول (أو الجداول) التي سيتم إدراجها في View.

لإنشاء View من متصفح الكائنات، يمكنك توسيع البند Databases ثم انقر بزر الأيمن على Views ثم اختر New View. وسيفتح مربع الحوار لإضافة الجداول:



يتم العمل على علية الحوار هذه، بنفس الطريقة التي استعرضناها في درس سابق

- لتحديد الجدول الذي سيتم استخدامه كمصدر للبيانات، انقر على الصفحة Tables.

- إذا أردت إضافة View


آخر، حدد الصفحة Views من الأعلى، ثم اختر كائن View الذي تود إدراجه.

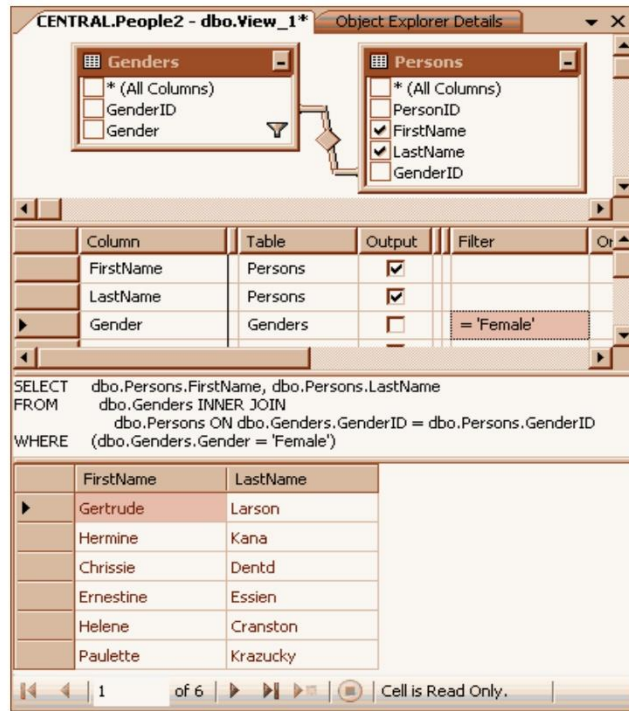
- إذا أردت استخدام إحدى الدوال لتوليد السجلات، يمكنك العثور عليها في الصفحة Functions.

بعد تحديد كائنات المصدر التي تود استخدامها لإنشاء View، يمكنك النقر نقرا مزدوجا عليها أو قم بتحديدتها ثم انقر على Add.

انقر على Close من مربع الحوار بعد إضافة جميع الكائنات التي تحتاجها لبناء View.

كما رأينا في درس بناء الاستعلام، يمكنك إنشاء صلة بين الجداول بسحب أي حقل من جدول وإسقاطه على أي حقل من جدول آخر، كذلك كتابة الاستعلام مباشرة وغير ذلك....

يمكنك في أي وقت اختبار ناتج View بتنفيذ الاستعلام بالنقر على زر التنفيذ . وبذلك تظهر في القسم السفلي من واجهة تصميم View نتائج الاستعلام. مثال:



من أهم مميزات كائنات Views هو إمكانية استخدامها مرارا وتكرارا. لتحقيق ذلك، يجب حفظ View في قاعدة البيانات. مثل باقي الكائنات في SQL Server يجب أن تمتلك View اسما خاصا بها، وأن يتم حفظها في قاعدة البيانات. لحفظ View في واجهة التصميم يمكن النقر على زر الحفظ Save من شريط الأدوات. يمكنك أيضا غلق واجهة التصميم ثم حفظ View بعد أن تتلقى حوار تأكيد خيار الحفظ، يتبع اسم View نفس القواعد التي رأيناها لتسمية الكائنات في SQL Server.

تطبيق عملي: إنشاء View في واجهة التصميم

- 1- في متصفح الكائنات، قم بتوسيع البند Databases ثم RealEstate2
- 2- بالزر الأيمن انقر على Views ثم اختر New View
- 3- في مربع الحوار Add Table انقر نقرا مزدوجا على PropertyTypes، Properties و Conditions

4- انقر على Close

5- من الجدول PropertyTypes، اسحب الحقل PropertyTypeID وأسقطه على الحقل Properties من الجدول PropertyTypeID

6- من الجدول Conditions، اسحب الحقل ConditionID وأسقطه على الحقل Properties من الجدول ConditionID

7- قم بتحديد الحقول التالية:

PropertyNumber, PropertyType, Condition, City, ZIPCode, Bedrooms, Bathrooms, FinishedBasement, Stories, YearBuilt, MarketValue

8- في قسم المعايير (Criteria)، انقر على PropertyType واضغط Tab 6 مرات، لتصل إلى عمود التصفية (Filter) واكتب Single Family

9- اضغط Tab، واكتب Townhouse في العمود Or...

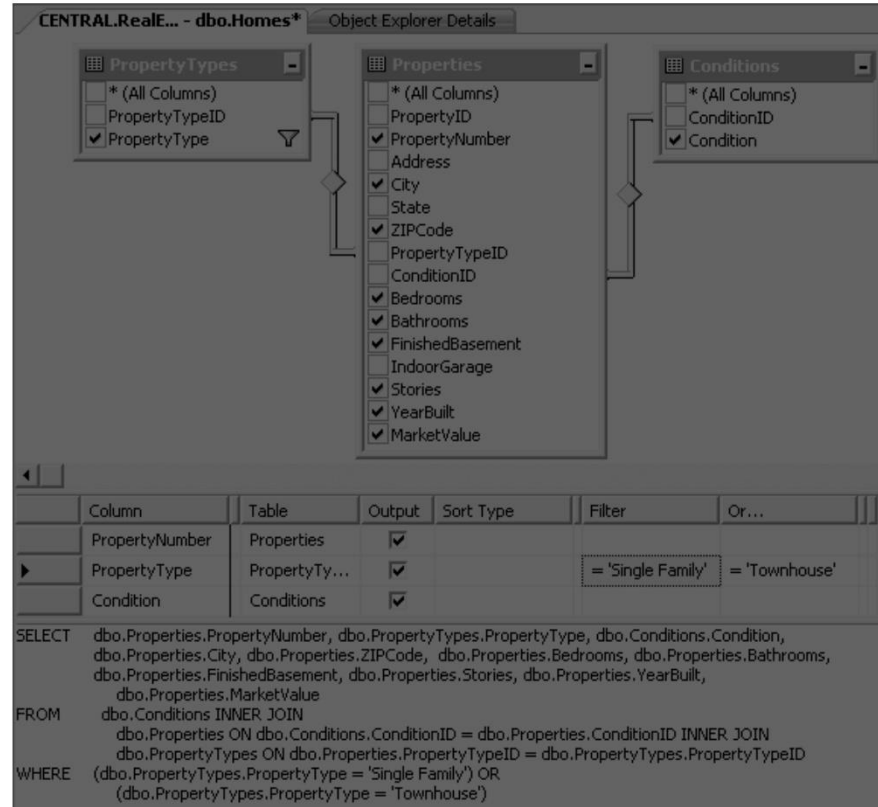
سنتبع في تسمية كائنات View في هذه الدروس نفس طريقة تسمية الكائنات الأخرى.

تصبح View بعد حفظها في قاعدة البيانات جزءاً منها، وسيتم إضافتها تحت البند الرئيسي Views بالاسم الذي تم حفظها به.

تطبيق عملي: حفظ كائن View

1- لحفظ View، انقر من شريط الأدوات Standard، على الزر Save

2- اكتب اسم View مثلاً: Homes، ثم انقر على OK



The screenshot shows the 'Object Explorer Details' window in SQL Server Enterprise Manager. It displays the 'dbo.Homes' view definition. The view is based on three tables: 'PropertyTypes', 'Properties', and 'Conditions'. The columns selected for the view are: PropertyNumber, PropertyType, Condition, City, ZIPCode, Bedrooms, Bathrooms, FinishedBasement, Stories, YearBuilt, and MarketValue. The view definition includes an INNER JOIN between 'Properties' and 'Conditions' on 'ConditionID', and another INNER JOIN between 'PropertyTypes' and 'Properties' on 'PropertyTypeID'. The WHERE clause filters for 'Single Family' or 'Townhouse' property types.

Column	Table	Output	Sort Type	Filter	Or...
PropertyNumber	Properties	<input checked="" type="checkbox"/>			
PropertyType	PropertyTy...	<input checked="" type="checkbox"/>		= 'Single Family'	= 'Townhouse'
Condition	Conditions	<input checked="" type="checkbox"/>			

```

SELECT dbo.Properties.PropertyNumber, dbo.PropertyTypes.PropertyType, dbo.Conditions.Condition,
dbo.Properties.City, dbo.Properties.ZIPCode, dbo.Properties.Bedrooms, dbo.Properties.Bathrooms,
dbo.Properties.FinishedBasement, dbo.Properties.Stories, dbo.Properties.YearBuilt,
dbo.Properties.MarketValue
FROM dbo.Conditions INNER JOIN
dbo.Properties ON dbo.Conditions.ConditionID = dbo.Properties.ConditionID INNER JOIN
dbo.PropertyTypes ON dbo.Properties.PropertyTypeID = dbo.PropertyTypes.PropertyTypeID
WHERE (dbo.PropertyTypes.PropertyType = 'Single Family') OR
(dbo.PropertyTypes.PropertyType = 'Townhouse')

```

3- أغلق واجهة تصميم View

4- من متصفح الكائنات، تحت البند RealEstate2، انقر بالزر الأيمن على Views ثم

اختر New View

5- من مربع الحوار Add Table، اختر الجداول PropertyTypes, Properties, Conditions

6- انقر على Close

7- من الجدول PropertyTypes، اسحب الحقل PropertyTypeID وأسقطه على الحقل PropertyTypeID من الجدول Properties

8- من الجدول Conditions، اسحب ConditionID وأسقطه على الحقل ConditionID من الجدول Properties

9- حدد من كل الجداول الحقول التالية:

PropertyNumber, PropertyType, Condition, City, ZIPCode, Bedrooms, Bathrooms, MarketValue

10- في قسم المعايير، انقر على PropertyType، اضغط على الزر Tab 3 مرات واضغط على الزر Space لإزالة علامة الاختيار من العمود Output

11- اضغط على 3 مرات على Tab، لتصل إلى عمود التصفية، واثم اكتب Condominiums واضغط Enter

12- من القائمة الرئيسية، انقر على File -> Save

13- اكتب اسم View مثلا Condos وانقر على OK

14- أغلق نافذة التصميم.

إنشاء View باستخدام SQL:

لإنشاء View برمجيا، يمكنك استخدام SQL وفقا للجملة التالية:

```
CREATE VIEW ViewName
AS
SELECT Statement
```

يمكن أن يقوم SQL Server Management Studio بتوليد هيكل SQL لإنشاء View. للقيام بذلك، أولا أنشئ إطار استعمال جديد، ثم من Template Explorer اسحب وأفنت البند Create View تحت البند الرئيسي View.

SQL إنشاء View يبدأ بعبارة CREATE VIEW متبوعا باسم مناسب، استخدم الكلمة AS بعد ذلك للدلالة على أن ما سيأتي من الاستعلام لتعريف محتوى View.

يتم تعريف View باستخدام عبارة SELECT، بإتباع نفس القواعد التي رأيناها خلال تحليل البيانات. مثال:

```
CREATE VIEW dbo.ListOfMen
AS
SELECT dbo.Genders.Gender,
       dbo.Persons.FirstName, dbo.Persons.LastName
FROM   dbo.Genders INNER JOIN dbo.Persons
ON     dbo.Genders.GenderID = dbo.Persons.GenderID
WHERE  (dbo.Genders.Gender = 'Male');
GO
```

بعد كتابة استعلام إنشاء View ثم تنفيذه بضغط F5 أو الزر Execute، أي بمجرد تنفيذ عبارة SQL، سيضاف اسم الكائن View بنفس الاسم الذي ورد في الاستعلام، وستتم إضافة إلى قائمة Views من قاعدة البيانات الحالية حتى لو لم يتم بحفظ الاستعلام.

تطبيق عملي: إنشاء View

1- تأكد من وجود قاعدة البيانات Yugo National Bank التي تم إنشاؤها في درس سابق.

2- من متصفح الكائنات، انقر بالزر الأيمن على YugoNationalBank وانقر على New Query

3- لإنشاء كائن View جديد، اكتب ما يلي:

```
Use YugoNationalBank;
GO
CREATE VIEW PayrollPreparation
AS
SELECT  dbo.Employees.EmployeeNumber,
        dbo.Employees.LastName + ', ' +
        dbo.Employees.FirstName AS [Full Name],
        dbo.Employees.HourlySalary,
        dbo.TimeSheets.TimeSheetCode, dbo.TimeSheets.Week1Monday,
        dbo.TimeSheets.Week1Tuesday, dbo.TimeSheets.Week1Wednesday,
        dbo.TimeSheets.Week1Thursday, dbo.TimeSheets.Week1Friday,
        dbo.TimeSheets.Week1Saturday, dbo.TimeSheets.Week1Sunday,
        dbo.TimeSheets.Week2Monday, dbo.TimeSheets.Week2Tuesday,
        dbo.TimeSheets.Week2Wednesday, dbo.TimeSheets.Week2Thursday,
        dbo.TimeSheets.Week2Friday, dbo.TimeSheets.Week2Saturday,
        dbo.TimeSheets.Week2Sunday
FROM    dbo.Employees INNER JOIN dbo.TimeSheets
ON      dbo.Employees.EmployeeNumber = dbo.TimeSheets.EmplNumber;
GO
```

4- قم بتنفيذ الاستعلام بالضغط F5

5- احذف محتوى نافذة الاستعلام.

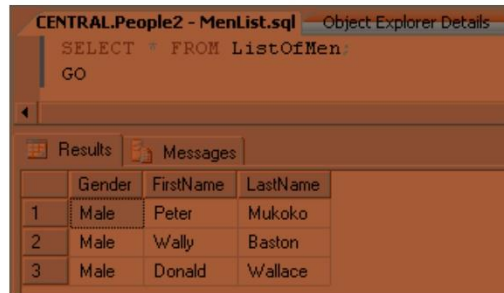
فتح أو تنفيذ View

بعد إنشاء كائن View، يمكنك عرضه بطريقتين:

- لمشاهدة هيكل View (الذي يمثل الجداول والعلاقات...)، انقر بالزر الأيمن من متصفح الكائنات على الكائن View ثم اختر Design
- لمشاهدة استعلام SQL الذي يمثل View، انقر بالزر الأيمن من متصفح الكائنات على الكائن View ثم اختر Edit.

تنفيذ كائن View يعني مشاهدة نتائجه، للقيام بذلك يمكنك:

- فتح نافذة استعلام مرتبطة بقاعدة البيانات التي تحتوي على View. ثم كتابة استعلام SELECT وفق نفس الصيغ والقواعد التي رأينا مع الجداول. مثال:



The screenshot shows a SQL query window titled 'CENTRAL.People2 - MenList.sql' with the following query: `SELECT * FROM ListOfMen;` and a 'GO' command. Below the query, the 'Results' tab is active, displaying a table with the following data:

	Gender	FirstName	LastName
1	Male	Peter	Mukoko
2	Male	Wally	Baston
3	Male	Donald	Wallace

- من متصفح الكائنات، بالزر الأيمن انقر على الكائن View ثم اختر Open View.

تطبيق عملي: تنفيذ View

- 1- من متصفح الكائنات، تحت البند RealEste2، قم بتوسيع Views، ثم انقر بالزر الأيمن على Condos ثم اختر Open View

PropertyNumber	Condition	City	ZIPCode	Bedrooms	Bathrooms	MarketValue
200417	Excellent	Germantown	NULL	2	1	215495,0000
NULL	Good	Rockville	NULL	2	2	436885,0000
288540	Good	Silver Spring	20906	1	1	242775,0000
NULL	Excellent	Takoma Park	NULL	2	NULL	360885,0000
NULL	Good	Rockville	NULL	1	1	418885,0000
811155	Good	Silver Spring	20906	1	1	252775,0000
NULL	Good	Rockville	NULL	1	1	420555,0000
644114	Good	Silver Spring	20906	2	2	258445,0000
*	NULL	NULL	NULL	NULL	NULL	NULL

2- أألق واجهة ناتج View

3- من متصفح الكائنات، انقر بالزر الأيمن على YugoNationalBank ثم اختر New Query

4- لتنفيذ كائن View اكتب استعلام كما يلي:

```
USE YugoNationalBank;
GO
SELECT PayrollPreparation.* FROM PayrollPreparation;
GO
```

5- لتنفيذ الكائن View، اضغط على F5

6- احذف محتوى نافذة الاستعلام.

صيانة كائنات View:

خصائص الكائن View:

يعتبر كائن View في SQL Server غرض (Object) كبقية الأغراض، للأجل ذلك يمكن مشاهدة محتواه، تعديله أو حذفه، ويملك كائن View خصائصه الخاصة. لمشاهدتها يمكنك النقر بالزر الأيمن على كائن View ثم انقر على Properties، لتشاهد نافذة الخصائص حاملة معلومات مختلفة، مثل اسم قاعدة البيانات وكائن View الذي ينتمي إليها، تاريخ إنشاء View، وغير ذلك...

تطبيق عملي: خصائص View

1- لإنشاء كائن View آخر، اكتب ما يلي:

```
-- =====
-- Database: YugoNationalBank
-- View:      TimeSheet
-- =====
USE YugoNationalBank
GO

IF object_id(N'TimeSheet', 'V') IS NOT NULL
    DROP VIEW dbo.TimeSheet
GO

CREATE VIEW dbo.TimeSheet
AS
SELECT EmplNumber, StartDate,
       Week1Monday, Week1Tuesday, Week1Wednesday,
       Week1Thursday, Week1Friday,
       Week1Saturday, Week1Sunday,
       Week2Monday, Week2Tuesday, Week2Wednesday,
       Week2Thursday, Week2Friday, Week2Saturday,
       Week2Sunday, Notes
FROM   TimeSheets;
```

2- اضغط F5 لتنفيذ الاستعلام

3- قم بحذف الاستعلام

4- من متصفح الكائنات، تحت البند YugoNationalBank، انقر بالزر الأيمن على Views واختر Refresh

5- ثم قم بتوسيع Views. انقر بالزر الأيمن على TimeSheet واختر Properties

6- اضغط على Esc لغلق نافذة الخصائص.

تعديل كائن View:

- بعد إنشاء كائن View، يمكنك تغييره أو تعديله حسب الحاجة، لأجل ذلك هناك خياران:
 - من متصفح الكائنات، انقر بالزر الأيمن على View، ثم اختر Design، يمكنك حذف أو إضافة الحقول من واجهة التصميم كما يمكنك تغيير أي من الخيارات في أي قسم من واجهة التصميم. بعد التعديلات على الكائن View قم بحفظها وأغلق واجهة التصميم.

- من متصفح الكائنات، انقر بالزر الأيمن على كائن View، اختر Edit، ثم قم بإجراء التعديلات اللازمة ولا تنسى حفظها.

- بطريقة أخرى من متصفح الكائنات، انقر بالزر الأيمن على الكائن View، ثم اختر:

Script View As -> ALTER To -> New Query Editor Window

الصيغة الأساسية لبرمجة لتعديل View هي:

```
ALTER VIEW ViewName
AS
SELECT Statement
```

تبدأ الاستعلام بعبارة ALTER VIEW متبوعة باسم كائن View الذي تود تعديله، بعد ذلك أضف الكلمة AS مشيراً إلى أن ما سيأتي من الاستعلام هو الذي سيغير محتوى View، ثم اكتب الاستعلام على النحو الذي تراه مناسباً. مثلاً، يمكنك إنشاء عبارة SELECT التي تتضمن تعديلات أو استعلام جديد تماماً.

يعرض كائن View الذي قمنا بإنشائه قائمة الذكور فقط من جدول الأشخاص، أضفنا أيضاً الحقل Gender، هذا الحقل زائد عن الحاجة، لأننا نعلم مسبقاً أن القائمة لا تضم سوى الذكور. مثال على تغيير View لإزالة (أو بالأحرى إقصاء) الحقل Gender من جدول الأشخاص:

```
ALTER VIEW dbo.ListOfMen
AS
SELECT dbo.Persons.FirstName, dbo.Persons.LastName
FROM dbo.Genders INNER JOIN dbo.Persons
ON dbo.Genders.GenderID = dbo.Persons.GenderID
WHERE (dbo.Genders.Gender = 'Male');
```

حذف كائن View:

يمكن حذف كائن View من قاعدة البيانات إذا لم تكن بحاجة إليه، يمكن ذلك بطرق مختلفة:

- من متصفح الكائنات ومن قاعدة بيانات، انقر بالزر الأيمن على اسم الكائن View الذي تود حذفه وانقر على Delete. وحينها ستتاح لك فرصة لتأكيد خيارك أو لتغيير رأيك.

- من متصفح الكائنات، انقر بالزر الأيمن الكائن View، ثم اختر:

Script View As -> DROP To New Query Editor Window

- بطريقة أخرى، افتح نافذة استعلام مرتبطة بقاعدة البيانات التي تحتوي على كائن View الذي تريد حذفه، من Template Explorer، من البند View، اسحب البند Drop View ثم أسقطه في إطار الاستعلام.

الصيغة العامة لحذف View هي:

```
DROP VIEW ViewName
```

في الجانب الأيمن لعبارة DROP VIEW أضف اسم الكائن View الذي تود حذفه، ثم قم بتنفيذ العبارة لحذف View. ولن تتلقى في هذه الحال أي تحذير من قبل SQL Server.

استخدام الكائن View:

إدخال البيانات باستخدام View:

تخيل أن لديك جدول الموظفين وتريد إنشاء قائمة بأسمائهم فقط. لأجل ذلك، يمكنك إنشاء View على النحو التالي:

```
CREATE VIEW dbo.EmployeesNames
AS
SELECT FirstName,
       LastName,
       LastName + ', ' + FirstName AS FullName FROM Persons;
GO
```

يستند كائن View السابق إلى جدول، حيث يمكنك القيام بإدخال البيانات إلى هذا الجدول من خلال كائن View، عوضاً عن إدخال البيانات مباشرة إلى الجدول. للقيام بذلك، يمكنك إتباع نفس القواعد التي استعرضنا في درس سابق. مثال:

```
INSERT INTO dbo.EmployeesNames(FirstName, LastName)
VALUES('Peter', 'Justice');
```

إدخال البيانات باستخدام View يعني إدخال البيانات في الجدول الأصل، وهذا يعني أن الجدول سيتم تحديثه تلقائياً. استناداً إلى هذه الميزة، يمكنك إنشاء View خصيصاً لتحديث البيانات في الجدول، بحيث تقوم بعرض الحقول التي يسمح بتحديثها فقط.

تطبيق عملي: إدخال البيانات باستخدام View

1- لإنشاء دالة سنستخدمها لاحقاً، أدخل الاستعلام التالي:

امثلة المحاضرة:

انشاء الجداول اللازمة للمثال وهي خاصة بتحديد صلاحيات أمنية في نظام حاسوبي معين:
جدول الصلاحيات: اضافة حذف تعديل ..

```
CREATE TABLE [dbo].[privilage](
[previd] [int] NOT NULL,
[prevname] [nvarchar](50) NULL,
CONSTRAINT [PK_privilage] PRIMARY KEY CLUSTERED
(
[previd] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
(جدول الأدوار (مدير - ...)
```

```
CREATE TABLE [dbo].[roles](
[roleid] [int] NOT NULL,
[rolename] [nvarchar](50) NULL,
CONSTRAINT [PK_roles] PRIMARY KEY CLUSTERED
(
[roleid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

جدول الربط بين الصلاحيات والأدوار:

```
CREATE TABLE [dbo].[prevrole](
[roleid] [int] NOT NULL,
[previd] [int] NOT NULL,
CONSTRAINT [PK_prevrole] PRIMARY KEY CLUSTERED
(
[roleid] ASC,
[previd] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

جدول يتضمن معلومات المستخدمين والأدوار الخاصة بهم:

```
CREATE TABLE [dbo].[sec](
[id] [int] NOT NULL,
[username] [nvarchar](50) NULL,
[pw] [nvarchar](50) NULL,
[role] [nvarchar](50) NULL,
CONSTRAINT [PK_sec] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
ALTER TABLE [dbo].[prevrole] WITH CHECK ADD CONSTRAINT [FK_prevrole_privilage] FOREIGN KEY([previd])
REFERENCES [dbo].[privilage] ([previd])
```

GO

```
ALTER TABLE [dbo].[prevrole] CHECK CONSTRAINT [FK_prevrole_privilage]
GO
```

```
ALTER TABLE [dbo].[prevrole] WITH CHECK ADD CONSTRAINT [FK_prevrole_roles] FOREIGN KEY([roleid])
REFERENCES [dbo].[roles] ([roleid])
GO
```

قادح لدى الحشر على جدول المستخدمين يتأكد من عدم وجود المستخدم بنفس الدور قبل الحشر:

```
create trigger [dbo].[new_ins_tri] on [dbo].[sec]
instead of insert
as
if (exists (select * from sec inner join inserted on
sec.username=inserted.username and sec.role=inserted.role))
print ('error in insert')
else
insert into sec select id,username,pw,role from inserted
```

انشاء منظار يربط بين جدول prevrole و privilege و جدول roles:

```
create view adminview as
SELECT dbo.privilage.prevname AS pname, dbo.roles.rolename AS rname, dbo.roles.roleid AS rid
FROM dbo.privilage INNER JOIN
dbo.prevrole ON dbo.privilage.previd = dbo.prevrole.previd INNER JOIN
dbo.roles ON dbo.roles.roleid = dbo.prevrole.roleid
```

لا يمكن الحشر على المنظار adminview كونه يربط بين أكثر من جدول وسوف يتم انشاء منظار بدل عملية الحشر يقوم القادح بما يلي:

التحقق من عدم وجود الصلاحية privilege مسبقا في الجدول privilege في حال وجود الصلاحية privilege مسبقا في الجدول يتم اخذ رقم الصلاحية في المتحول previd والا يتم حشر الصلاحية اولا في جدول privilege ثم يتم التحقق من ان ال role غير موجود وفي حال كان موجود يتم الحصول عليه من جدول role وحفظه في المتحول @roleid وبعد عمليات التحقق والحشر يكون قد تم الحصول على رقم privilege ورقم role ويتم حشر المعلومات في جدول prevrole الذي يربط بين الدور والصلاحية

```
ALTER trigger [dbo].[ins_trig] on [dbo].[adminview] instead of insert
as
declare @previd int
if exists (select * from privilage where prevname in (select pname from
inserted))
select @previd=previd from privilage where prevname=(select pname from inserted)
else
begin
select @previd=isnull(max(previd)+1,1) from privilage
insert into privilage values(@previd,(select pname from inserted))
end
declare @roleid int
if exists (select * from roles where rolename in (select rname from
inserted))
select @roleid=roleid from roles where rolename=(select rname from inserted)
else
begin
select @roleid=isnull(max(roleid)+1,1) from roles
insert into roles values(@roleid,(select rname from inserted))
end
```

```
insert into prevrole values(@roleid,@previd)
```

انشاء قادم يقوم باستبدال عملية الحذف على المنظار adminview كونه يربط بين أكثر من جدول وبالتالي لا يمكن لعملية الحذف من المنظار تحديد الجدول المطلوب الحذف منه لذلك نقوم بانشاء قادم لاستبدال عملية الحذف

```
create trigger [dbo].[deltrig] on [dbo].[adminview] instead of delete
as
delete from prevrole where roleid in(select rid from deleted)
```

قادم يقوم بالتحقق من عملية الحذف من جدول sec الذي يتضمن معلومات المستخدمين يقوم القادم بمنع حذف المستخدم باسم admin والتأكد من وجود سطرين على الأقل في جدول sec

```
create trigger [dbo].[del_tri] on [dbo].[sec] instead of delete as
declare @id int
declare @usname nvarchar(50)
declare @c int
select @id =id,@usname=username from deleted
select @c=isnull(count(*),0) from sec where role ='admin'
if @c<=2
print ('error @c less than 2')

else if @usname='admin'
print ('error')
else
delete from sec where id =@id
```