**Introduction:**  
In this article we are going to discuss about the Views in the sql server 2005 version. It is one of the important elements in the sql server; it eliminates the difficulties in the business process. There are lots of significant improvements in the view especially in the sql server 2005 version. They have introduced the indexed views and INSTEAD of Trigger on the permanent view. Let us discuss about the views with more practical way.  
 **Views:**  
The view is a virtual table, which can have the multiple columns from the one or more table. It can be used like the normal table. Normally view cannot store the data permanently in the table. When we create the view it stores the view definition schema as object under the concern database.  
  
Let us see the syntax of the create view  
  
CREATE VIEW View Name [Alias name1, name2,]  
  
WITH ENCRYPTION  
  
WITH SCHEMA BINDING  
  
AS  
  
SELECT statement [WITH CHECK OPTION]  
  
The create view can be created with the view name and the alias can be given in the view name parameter parenthesis. The view schema can be stored in the encrypted format. Here is an option like SCHEMA BINDING; this is an important mile stone in the view to allow the developers to create the permanent view.  
 **When to use VIEW?**  
When you have complex queries, that use many places in the stored procedures or functions, etc..,  
  
It will be used as security mechanism in the web applications. When we use the original table in the web applications the hackers may drop the table. That time the original data will be persist in the table.  
  
When you want to hide the particular columns to the specific people then we can create the specialized view.  
 **Encrypted View:**  
The definition of schema will be encrypted and stored as object in the database. This can be done using the ENCRYPTION option in the view creation.  
  
IF OBJECT\_ID('[DBO].Vw\_SqlObjects\_Encrypted') IS NOT NULL  
BEGIN  
   DROP VIEW [DBO].Vw\_SqlObjects\_Encrypted  
   PRINT '<< [DBO].Vw\_SqlObjects\_Encrypted View dropped >>'  
END  
GO  
CREATE VIEW [DBO].Vw\_SqlObjects\_Encrypted  
WITH ENCRYPTION  
AS  
   SELECT  
       O.Object\_ID  
      ,O.Name  
      ,'Type' = CASE O.type WHEN 'S' THEN 'Scalar Functions'  
                          WHEN 'F' THEN 'Functions'  
                          WHEN 'V' THEN 'Views'  
                          WHEN 'PK' THEN 'Primary keys'  
                          WHEN 'TR' THEN 'Triggers'  
                          WHEN 'P' THEN 'Procedures'  
                          WHEN 'U' THEN 'User Defined Functions'  
                          WHEN 'TF' THEN 'Table Valued Functions'  
                          WHEN 'IF' THEN 'Inline Functions' END  
      ,O.create\_date  
      ,O.modify\_date  
      ,CASE WHEN SC.encrypted = 0 THEN 'No' ELSE 'Yes' END AS [IsEncrypted]  
      ,SC.text  
  FROM  
      SYS.OBJECTS O  
  INNER JOIN  
      SYSCOMMENTS SC ON SC.id = O.object\_id   
 GO  
IF OBJECT\_ID('[DBO].Vw\_SqlObjects\_Encrypted') IS NOT NULL  
BEGIN  
   PRINT '<< [DBO].Vw\_SqlObjects\_Encrypted View created >>'  
END  
GO  
  
Now if you want to see the view schema definition for the above view is not possible. We have stored in the encrypted format. This is a significant option to hide the important calculations inside the view from the others.  
  
In case of any alter in the view must be stored externally somewhere else.   
  
SELECT text FROM SYSCOMMENTS WHERE id = OBJECT\_ID('[DBO].Vw\_SqlObjects\_Encrypted')

SELECT definition FROM SYS.sql\_modules WHERE object\_id = OBJECT\_ID('[DBO].Vw\_SqlObjects\_Encrypted')

sp\_helptext Vw\_SqlObjects\_Encrypted  
  
If you execute the above queries then it will say like view is encrypted.  
  
There are three types of views in the sql server 2005.  
  
They are

1. Normal or Standard view
2. Indexed or permanent view
3. Partitioned view

**Normal or Standard view:**  
This view is most frequently used by the developers. When create the view the schema will be stored an object in the database. When we retrieve the content from this virtual table, it will be executed the schema and stored the data from the parent table.  
  
Here if you have the result from the same table then it can be updated and inserted. The deleted row will be reflected in the original table.  
  
USE [Northwind]  
GO  
IF OBJECT\_ID('[DBO].vw\_ViewProducts','V') IS NOT NULL  
BEGIN  
  DROP VIEW [DBO].vw\_ViewProducts  
  PRINT '<< [DBO].vw\_ViewProducts view dropped.. >>'  
END  
GO  
CREATE VIEW [DBO].vw\_ViewProducts  
AS  
 SELECT   
    ProductID,ProductName,SupplierID,CategoryID,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder,ReorderLevel,Discontinued  
 FROM Products  
GO  
IF OBJECT\_ID('[DBO].vw\_ViewProducts','V') IS NOT NULL  
BEGIN  
  PRINT '<< [DBO].vw\_ViewProducts view created.. >>'  
END  
GO  
--O/P  
SELECT \* FROM [DBO].vw\_ViewProducts  
--INSERT  
INSERT INTO [DBO].vw\_ViewProducts(ProductName,SupplierID,CategoryID,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder,ReorderLevel,Discontinued)  
VALUES('Test View',1,2,'100 per bag',25.45,89,57,15,0)  
--DELETE  
DELETE FROM [DBO].vw\_ViewProducts WHERE ProductID = 81  
  
Here you can do the DML operations in the view when you have only one table.   
 **Indexed views:**  
The indexed or permanent view is one of the new features introduced in the sql server 2005 version. We have seen that the view only store the schema definition and it will get execute and load the data into the virtual table at the time of view used. But this view creates the permanent view and we can create the indexes on the table. It allows us to create the instead of trigger.  
  
The indexed view can be created with the WITH SCHEMA BINDING option while creating the view.  
  
The indexed view has some restrictions like cannot use the TOP, DISTINCT, UNION, ORDER BY and aggregate functions.  
  
It allows us to use the GROUP BY statement but we cannot use COUNT statement. Instead of that COUNT\_BIG statement can be used.  
  
IF EXISTS(SELECT OBJECT\_ID FROM SYS.OBJECTS WHERE OBJECT\_ID = OBJECT\_ID(N'[DBO].Vw\_Product\_Sales\_Report',N'V'))  
BEGIN  
  DROP VIEW [DBO].Vw\_Product\_Sales\_Report  
  PRINT '<< [DBO].Vw\_Product\_Sales\_Report view dropped >>'  
END  
GO  
CREATE VIEW [DBO].Vw\_Product\_Sales\_Report  
WITH SCHEMABINDING  
AS  
  SELECT   
      O.OrderID  
     ,C.CustomerID  
     ,C.CompanyName  
     ,C.Address+', '+C.City AS [Customer Address]  
     ,OD.ProductID  
     ,P.ProductName  
     ,OD.UnitPrice  
     ,OD.Quantity  
     ,(OD.UnitPrice \* OD.Quantity) AS [Total]  
     ,(OD.UnitPrice \* OD.Quantity) \* OD.Discount/100 AS [Discount]  
   FROM  
     [DBO].Orders O (NOLOCK)  
   INNER JOIN [DBO]."Order Details" OD (NOLOCK) ON OD.OrderID = O.OrderID  
   INNER JOIN [DBO].Customers C (NOLOCK) ON C.CustomerID = O.CustomerID  
   INNER JOIN [DBO].Products P (NOLOCK) ON P.ProductID = OD.ProductID  
GO  
IF EXISTS(SELECT OBJECT\_ID FROM SYS.OBJECTS WHERE OBJECT\_ID = OBJECT\_ID(N'[DBO].Vw\_Product\_Sales\_Report',N'V'))  
BEGIN  
  PRINT '<< [DBO].Vw\_Product\_Sales\_Report view created >>'  
END  
GO  
  
Here the  indexed view has created. When you retrieve the data from this table, it will execute like normal table.  
  
There are some retrictions while creating this indexed view like the name of the view must be two part name and we cannot use select \* in the view schema defintion.  
  
Normally view cannot have the triggers but from the sql server 2005 onwards We can create the Instead of trigger on the instead of trigger.  
 **Partitioned Views:**  
The partitioned view and its execution is like normal view. It will work across the database and across the server.  
  
There are two types of Partitioned views. They are

1. Local Partitioned View
2. Global Partitioned View

**1. Local Partitioned View:**  
The local partitioned view can be created within same server but different database.  
  
The view schema definition will be stored in the executed database. But when we try to retrieve the data from the table, it has to execute the schema definition internally and load the data.  
  
Let us see an example.  
  
USE [Northwind]  
GO  
CREATE TABLE EmployeeList  
(  
  iEmployeeID INT IDENTITY(1,1),  
  vFirstName VARCHAR(25) NOT NULL,  
  vLastName VARCHAR(25) NOT NULL,  
  iDeptID INT,  
  vAddress VARCHAR(25) NOT NULL,  
  vCity VARCHAR(25) NOT NULL,  
  vState VARCHAR(25) NOT NULL,  
  vCountry VARCHAR(25) NOT NULL,  
)  
GO  
USE [Master]  
GO  
CREATE TABLE Department  
(  
  iDeptID INT IDENTITY(1,1) PRIMARY KEY,  
  vDeptName VARCHAR(50),  
  vDeptDesc VARCHAR(25),  
  vDeptAddedBy VARCHAR(50),  
  vPostedDate DATETIME DEFAULT GETDATE()  
)  
GO  
--SELECT \* FROM Department  
USE [Northwind]  
GO  
IF OBJECT\_ID('[DBO].vw\_LocalPartion\_View','V') IS NOT NULL  
BEGIN  
  DROP VIEW [DBO].vw\_LocalPartion\_View  
  PRINT '[DBO].vw\_LocalPartion\_View view dropped...'  
END  
GO  
CREATE VIEW [DBO].vw\_LocalPartion\_View  
AS  
SELECT E.iEmployeeID,E.vFirstName+SPACE(1)+E.vLastName AS [Name],  
       D.vDeptName,E.vAddress,E.vCity,E.vState  
FROM EmployeeList E  
--INNER JOIN Master..Department D ON D.iDeptID = E.iDeptID --Either one of the way will be used.  
INNER JOIN Master.dbo.Department D ON D.iDeptID = E.iDeptID  
GO  
IF OBJECT\_ID('[DBO].vw\_LocalPartion\_View','V') IS NOT NULL  
BEGIN  
  PRINT '[DBO].vw\_LocalPartion\_View view created...'  
END  
GO  
--O/p  
SELECT \* FROM [DBO].vw\_LocalPartion\_View   
 **2. Global Partitioned View**  
The global Partitionedview will work across the server. The view can be created to join the table across the server.  
  
The accessing format will be like this.  
  
[Server Name].  Database Name. Table Name  
  
When we execute the view if it is not linked with the current server then it will ask us to link the external server.  
  
The following system stored procedure will be used to link the server.  
  
sp\_addlinkedserver 'Server name'  
  
The following system catalog table is used to see the list of linked servers.  
  
SELECT \* FROM SYS.SERVERS  
 **INSTEAD OF Triggers on the Indexed View**  
Normally the triggers cannot be created on the view. But sql server 2005 onwards we can create the INSTEAD OF trigger on the indexed views.  
  
USE [Northwind]  
GO  
IF OBJECT\_ID('[DBO].[VW\_Trigger\_Example') IS NOT NULL  
BEGIN  
   DROP VIEW [DBO].[VW\_Trigger\_Example]  
   PRINT '[DBO].[VW\_Trigger\_Example view dropped..'  
END  
GO  
CREATE VIEW [DBO].[VW\_Trigger\_Example]  
WITH SCHEMABINDING  
AS  
  SELECT P.ProductID,P.ProductName,P.SupplierID,  
         OD.OrderID,OD.UnitPrice,OD.Quantity  
  FROM [DBO].Products P  
  INNER JOIN [DBO].[Order Details] OD ON OD.ProductID = P.ProductID  
GO  
IF OBJECT\_ID('[DBO].[VW\_Trigger\_Example') IS NOT NULL  
BEGIN  
   PRINT '[DBO].[VW\_Trigger\_Example view created..'  
END  
GO  
--SELECT \* FROM VW\_Trigger\_Example  
IF OBJECT\_ID('[DBO].Tr\_Delete\_TriggerExample','TR') IS NOT NULL  
BEGIN  
  DROP TRIGGER [DBO].Tr\_Delete\_TriggerExample  
  PRINT '[DBO].Tr\_Delete\_TriggerExample trigger dropped..'  
END  
GO  
CREATE TRIGGER [DBO].Tr\_Delete\_TriggerExample  
ON [DBO].VW\_Trigger\_Example  
INSTEAD OF DELETE  
AS  
BEGIN  
   PRINT '----------------------------------------'  
   PRINT 'This is an example of INSTEAD OF Trigger'  
   PRINT '----------------------------------------'  
   SELECT TOP 1 \* FROM DELETED  
END  
GO  
IF OBJECT\_ID('[DBO].Tr\_Delete\_TriggerExample','TR') IS NOT NULL  
BEGIN  
  PRINT '[DBO].Tr\_Delete\_TriggerExample trigger created..'  
END  
GO  
--O/P  
--SELECT \* FROM [DBO].[VW\_Trigger\_Example] WHERE ProductID = 11  
DELETE FROM [DBO].[VW\_Trigger\_Example] WHERE ProductID=11  
 **How to view the Created Views?**  
There are few ways to view the scehema definition of the created views.  
  
SP\_HELPTEXT vw\_LocalPartion\_View  
SELECT id,text FROM SYSCOMMENTS WHERE id = OBJECT\_ID('[DBO].vw\_LocalPartion\_View')  
SELECT object\_id,definition FROM SYS.SQL\_MODULES WHERE OBJECT\_ID = OBJECT\_ID('[DBO].vw\_LocalPartion\_View')  
 **How to drop the View?**  
If you want to drop the view then you can use the following statement. When you drop the table underlying view will not be deleted. But if you run that view it will thrown an error.  
  
DROP VIEW VIEW\_NAME  
 **How to alter the view?**  
If you want to do changes in the created views then you can alter the view whatever you want to view the same view name.  
  
ALTER VIEW VIEW\_NAME

AS

SELECT [Columns List]....