/\*

What is an SQL view?

A view can be thought of as either a virtual table or a stored query which is stored in the database as a SELECT statement

A user can use this virtual table just as he/she would the same way a table

Views are created by defining the SELECT statement and then prefixing that statement with the CREATE VIEW command

The table that provide the data are called base tables

The view itself does not hold data. Since views are 'formed' from a base table, they can be queried, updated, and dropped

The view is essentially a dynamic SELECT query, and if any changes are made to the originating table(s),

These changes will be reflected in the SQL VIEW automatically

Some of the ways and benefits of using a view are as follows:

• Restrict a user to specific rows in a table by filtering row data

• Restrict a user to specific columns by filtering column data

• Join columns from multiple tables so that they look like a single table.

• Hide complexity of the code

• Use as a security mechanism by giving permissions set on the view instead of the underlying tables.

\*/

Use AdventureWorks2012

Go

--Base table

SELECT [BooksID]

, [BookTitle]

, [BookAuthor]

, [BookQuantity]

, [SoldDate]

FROM [AdventureWorks2012]. [dbo].[Books]

--Restrict a user to specific columns in a table by filtering columns

--Write the SELECT statement followed by the CREATE VIEW command

Use AdventureWorks2012

Go

CREATE VIEW vwBooks

AS

SELECT [BooksID]

,[BookTitle]

,[BookAuthor]

FROM [AdventureWorks2012].[dbo].[Books] --<< note we have omitted various fields

--Viewing data from the view, just like the table

Select \* from vwBooks

--Join columns from multiple tables so that they look like a single table.

USE AdventureWorks2012

GO

CREATE VIEW vwEmpEmail

AS

SELECT

Person.Person.FirstName, Person.Person.LastName, HumanResources.EmployeePayHistory.Rate, Person.EmailAddress.EmailAddress

FROM

Person. Person

INNER JOIN

HumanResources.EmployeePayHistory

ON Person.Person.BusinessEntityID = HumanResources.EmployeePayHistory.BusinessEntityID INNER JOIN

Person.EmailAddress

ON Person.Person.BusinessEntityID = Person.EmailAddress.BusinessEntityID

--Viewing data from the view, just like the table

Select \* from vwEmpEmail

--Deleting and updating of views are propagated to the base table and vice versa

--Backup table

Select \* Into [dbo].[Books\_040215]

FROM [dbo].[Books]

--(20 row(s) affected)

SELECT \*

FROM [dbo].[Books]

--(20 row(s) affected)

Select \*

from [dbo].[vwBooks]

--(20 row(s) affected)

--Deleting from base table two rows and it's propagated to view

Delete

From [dbo].[vwBooks]

Where BooksID IN (1002,1003)--<< Must use the where clause, otherwise all data will be deleted

--Base table

SELECT \*

FROM [dbo].[Books]

--18 rows

Select \*

From [dbo].[vwBooks]

--18 rows

--Insert rows in VIEW and it’s propagated to base table

Insert Into dbo.Books

Values

('Pride and Prejudice', 'Jane Austen', 32, '2015-03-10 00:00:00.000'),

('The Lord of the Rings','JRR ',555,'2015-03-10 00:00:00.000')

--Base table

SELECT \*

FROM [dbo].[Books]

where BookTitle IN ('Pride and Prejudice', 'The Lord of the Rings')

--View

Select \*

From [dbo].[vwBooks]

where BookTitle IN ('Pride and Prejudice', 'The Lord of the Rings')

--PERMISSION TO VIEWS

--the DBA can control what user can do with a view, such as select, update, insert or delete.

--COURSE 2 (SECURITY AND PERMISSIONS)