View and Function

.NET CORE

A View is a way to create a SQL table virtually for a specific purpose, such as to present data to a user in a way that safeguards the data from malicious intent.

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15

SQL – Computed Columns

https://docs.microsoft.com/en-us/sql/relational-databases/tables/specify-computed-columns-in-a-table?view=sql-server-ver15

A Computed Column is a virtual column whose value is based on some computation done on other columns within the table. It is not physically stored in the table unless the column is marked *PERSISTED*.

A computed column expression can use data from other columns to calculate a value for the column to which it belongs.

Use the keyword AS to designate a column as a Computed Column.

```
CREATE TABLE dbo.Products
(ProductID int IDENTITY (1,1) NOT
NULL,
QtyAvailable smallint,
UnitPrice money,
InventoryValue
AS
QtyAvailable * UnitPrice);
```

SQL – Computed Tables (Views)

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15

A Computed Table is a virtual table whose contents are defined by a query. A view can be used for:

- To focus, simplify, and customize the perception each user has of the database.
- As a security mechanism by allowing users to access data through the view, without granting the users permissions to directly access the underlying base tables.
- To provide a backward compatible interface to emulate a table whose schema has changed.

```
ALTER TABLE Poke.Pokemon ADD
    Active BIT NOT NULL DEFAULT 1;

GO
CREATE VIEW Poke.ActivePokemon AS
    SELECT * FROM Poke.Pokemon WHERE Active = 1;
GO
```

View – WITH SCHEMABINDING

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15#arguments
https://www.tutorialspoint.com/sql/sql-using-views.htm

When ***SCHEMABINDING*** is specified:

- the base table(s) cannot be modified in a way that would affect the view definition.
- The view definition itself must first be modified or dropped to remove dependencies on the table that is to be modified.
- the ***SELECT*** statement must include the two-part names (schema.object) of tables, views, or user-defined functions that are referenced.
- All referenced objects must be in the same database.

```
CREATE VIEW view_name
WITH SCHEMABINDING
AS
SELECT column1, column2...
FROM table_name
WHERE [condition];
```

View – WITH SCHEMABINDING

WITH SCHEMABINDING sets up a "hard" reference from the view to the table. The view prevents any changes to that table that would “break” the view's query

```
GO
CREATE VIEW Poke.WeirdView WITH SCHEMABINDING AS
    SELECT PokemonId * 2 AS PokemonId, Name + '!' AS Name
    FROM Poke.Pokemon;
GO
DROP VIEW Poke.WeirdView;
DROP TABLE Poke.Pokemon;
```

```
SELECT * FROM Poke.WeirdView;
DELETE FROM Poke.WeirdView WHERE PokemonId = 2000;
UPDATE Poke.WeirdView SET Name = 'Charmander';
```