

---

Helps simplify database structure to get data easier

- Select statement designed to act as a table
- Uses:
  - Simplifies complex table structure
    - Helps simplify filtering of data
    - Makes complex joins easier to use
  - Enhances security
    - Can give permission to the view but not the underlining data
  - Potentially increased performance
    - Can create indexed view, where data is written out to disk

Put **create view** on top of select statement

Once a view is created you can use it like a table and joins it to other tables

Give view name and view data from view as a table

Table gets built in memory

---

## View Limitations

- Max 1024 columns
- No order by statement unless using TOP
- Single query returning a single table
- Columns must be directly referenced
  - Not AVG, COUNT, SUM, MIN, MAX
- Updates must be for one table only
- Cannot use SELECT ... INTO
- Cannot reference a temp table or table variable

## Limits on modifying view data

- Cannot modify data in multiple tables
- Cannot modify if using an aggregate function

## View Best Practices

- Use 2-part naming

- Always start view name with lower case **v** to denote it's a view and not a table
- Don't join views
- Use good query writing
  - Don't do select \*
  - Only returned required data

## View Syntax

Basic:

CREATE VIEW <ViewName>

As

<SELECT statement>

Complete Syntax:

```
-- Create or Alter view
{ CREATE | ALTER } VIEW <name>
    [WITH <options>]
AS
-- SELECT statement
[ WITH CHECK OPTION ]

-- Drop view
DROP VIEW <name>;
```

---

## View Options

- ENCRYPTION
  - Encrypts the definition, preventing all users from accessing the script
  - Save your script if using this
  - Issues

- Nobody can access it
- Easily hacked
- Only encrypts on last statement if you alter it, then it's not encrypted
- SCHEMABINDING
  - Underlying tables cannot be modified in a way that would impact the view
    - Required for indexed views
- View\_METADATA
  - Enables external APIs to browse metadata
- CHECK Option
  - Specified at the end
  - Prevents modifications that would cause data to leave the view

- Add a new query

**SQL**

```
USE AdventureWorks2012
go

Select c.firstName,
c.lastName, c.EmailPromotion, e.HireDate
from Person.Person c
inner join humanresources.employee e
on c.BusinessEntityID=e.BusinessEntityID
```

**RESULT**

Results Messages

	firstName	lastName	EmailPromotion	HireDate
1	Margie	Shoop	2	2003-02-05
2	Rebecca	Laszlo	1	2003-01-30
3	Annik	Stahl	0	2003-01-18
4	Suchitra	Mohan	2	2003-03-20
5	Brandon	Heidepriem	1	2003-03-12
6	Jose	Lugo	2	2003-03-14
7	Chris	Okelberry	0	2003-04-08

7M6IOSH6U3\SQLSERVER20... | WIN-A7M6IOSH6U3\testus... | AdventureWorks2012 | 00:00:00 | 290 rows

- Now let's create a view out of this

**SQL**

```
CREATE VIEW HumanResources.vEmployeeList2
as
Select c.firstName,
c.lastName, c.EmailPromotion, e.HireDate
from Person.Person c
inner join humanresources.employee e
on c.BusinessEntityID=e.BusinessEntityID
```

**RESULT**

Command(s) completed successfully.

- Name it with v prefix to denote a view
- Now lets run a query using our new view

**SQL**

```
select *
from HumanResources.vEmployeeList2
```

**RESULT**

	firstName	lastName	EmailPromotion	HireDate
1	Margie	Shoop	2	2003-02-05
2	Rebecca	Laszlo	1	2003-01-30
3	Annik	Stahl	0	2003-01-18
4	Suchitra	Mohan	2	2003-03-20
5	Brandon	Heidepriem	1	2003-03-12
6	Jose	Lugo	2	2003-03-14
7	Chris	Okelberry	0	2003-04-08

7M6IOSH6U3\SQLSERVER20... | WIN-A7M6IOSH6U3\testus... | AdventureWorks2012 | 00:00:00 | 290 rows

- You can filter and sort data

**SQL**

```
select LastName + ',' + FirstName as 'FullName',
EmailPromotion, HireDate
from HumanResources.vEmployeeList2
```

**RESULT**

	FullName	EmailPromotion	HireDate
1	Sánchez, Ken	0	2003-02-15
2	Duffy, Terri	1	2002-03-03
3	Tamburello, Roberto	0	2001-12-12
4	Walters, Rob	0	2002-01-05
5	Erickson, Gail	0	2002-02-06
6	Goldberg, Jossef	0	2002-02-24

Query... | WIN-A7M6IOSH6U3\SQLSERVER20... | WIN-A7M6IOSH6U3\testus... | AdventureWorks2012 | 00:00:00 | 290 rows

- Add a where clause

**HTML**

```
select LastName + ',' + FirstName as 'FullName',
EmailPromotion, HireDate
from HumanResources.vEmployeeList2
where LastName like 'H%';
```

**CSS**

Results Messages

	FullName	EmailPromotion	HireDate
1	Hampadounsataya,Sariya	0	2003-01-13
2	Hamilton,James	0	2003-03-07
3	Heidepriem,Brandon	1	2003-03-12
4	Hill,Andrew	1	2003-03-26
5	Higa,Sidney	0	2002-03-05
6	Hartwig,Doris	0	2002-04-11

Query e... | WIN-A7M6IOSH6U3\SQLSERVER20... | WIN-A7M6IOSH6U3\testus... | AdventureWorks2012 | 00:00:00 | 23 rows

- Add order by
  - Order by is not allowed in view definition because it can be added to query calling the view and might cause it to sort twice

**SQL**

```
select LastName + ',' + FirstName as 'FullName',
EmailPromotion, HireDate
from HumanResources.vEmployeeList2
where LastName like 'H%'
order by lastname;
```

**RESULT**

Results Messages

	FullName	EmailPromotion	HireDate
1	Hagens,Erin	0	2004-03-03
2	Hall,Don	0	2003-03-17
3	Hamilton,David	1	2003-02-04
4	Hamilton,James	0	2003-03-07
5	Hampadounsataya,Sariya	0	2003-01-13
6	Hamington,Mark	0	2003-02-16

Query e... | WIN-A7M6IOSH6U3\SQLSERVER20... | WIN-A7M6IOSH6U3\testus... | AdventureWorks2012 | 00:00:00 | 23 rows

## SchemaBinding

**SQL**

```
CREATE VIEW Humanresources.vEmployeeList3 WITH SCHEMABINDING
as
```

```
select c.firstname  
from Person.Person c  
go
```

**RESULT**

Command(s) completed successfully.

With schemabinding must include schema name along with the table

If try to drop the table

**SQL**

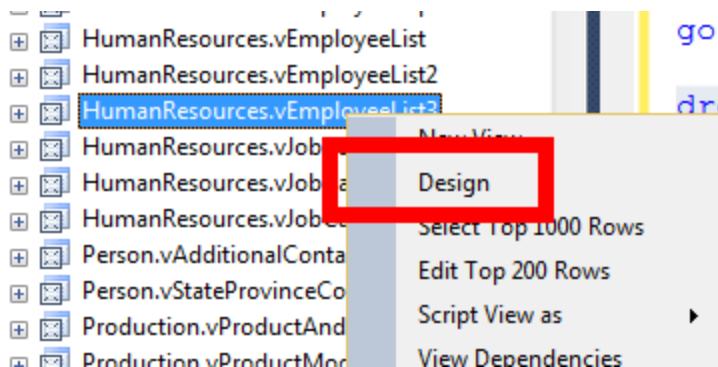
```
drop table Person.Person
```

**RESULT**

Msg 3726, Level 16, State 1, Line 1  
Could not drop object 'Person.Person' because it is referenced by a FOREIGN KEY constraint.

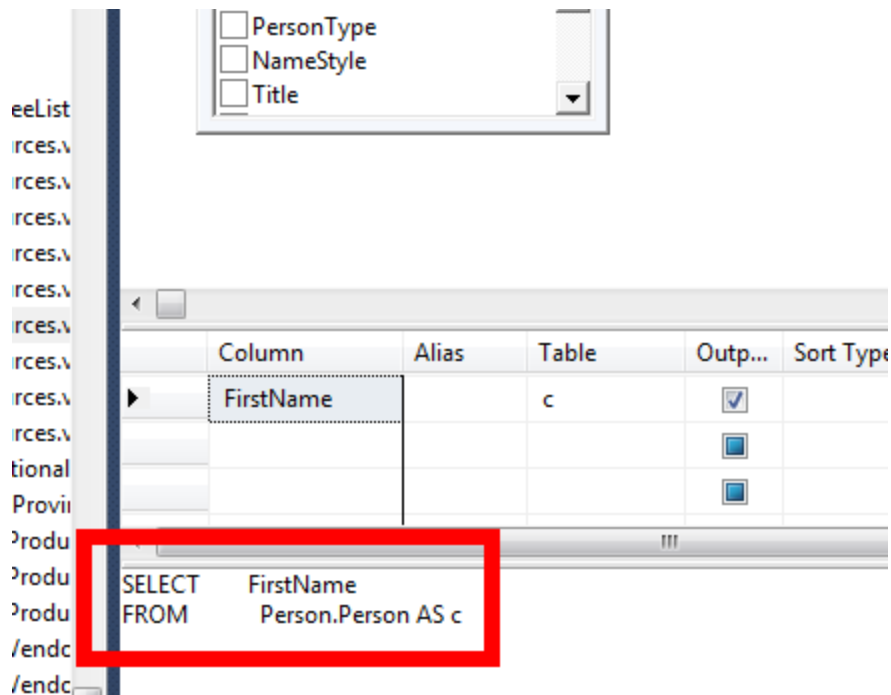
It prevents it

Right click on view and choose design and you can see it



Copy and paste to make new view

Create a view on a view



## Encryption

## Create another view

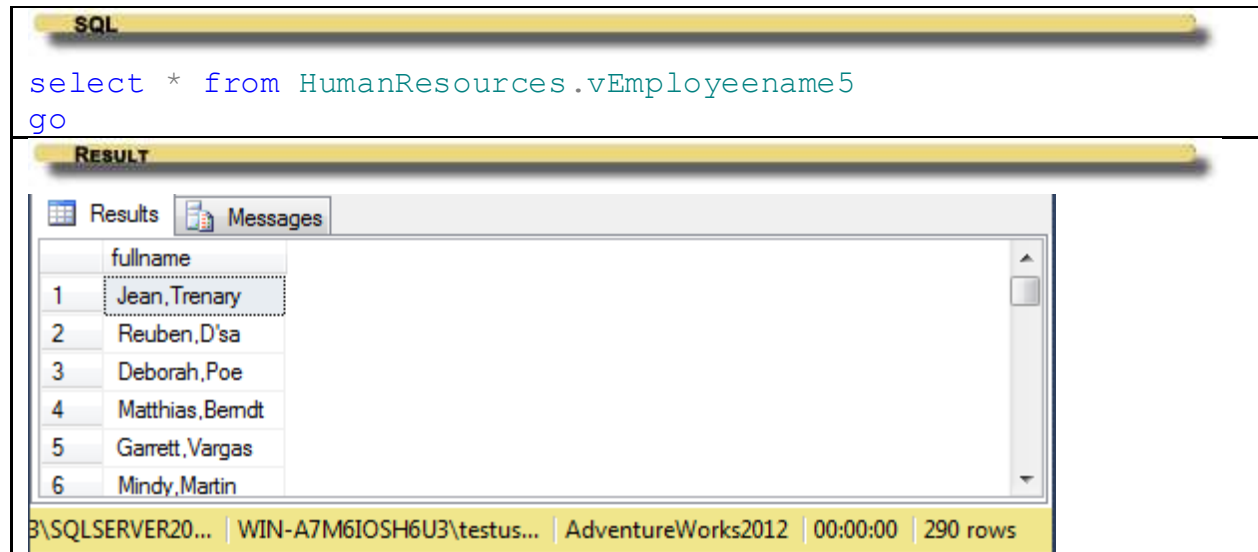
Refresh the views and look at the design(its grayed out)

```
SQL
CREATE VIEW Humanresources.vEmployeeList4
With encryption
as
select c.firstname
from Person.Person c
go
```

## Nesting view

## Creating a view from pieces of another view

```
SQL
create view humanresources.vEmployeeName5
as
select v.firstname + ',' + v.lastname as 'fullname'
from humanresources.vEmployeeList2 v
```



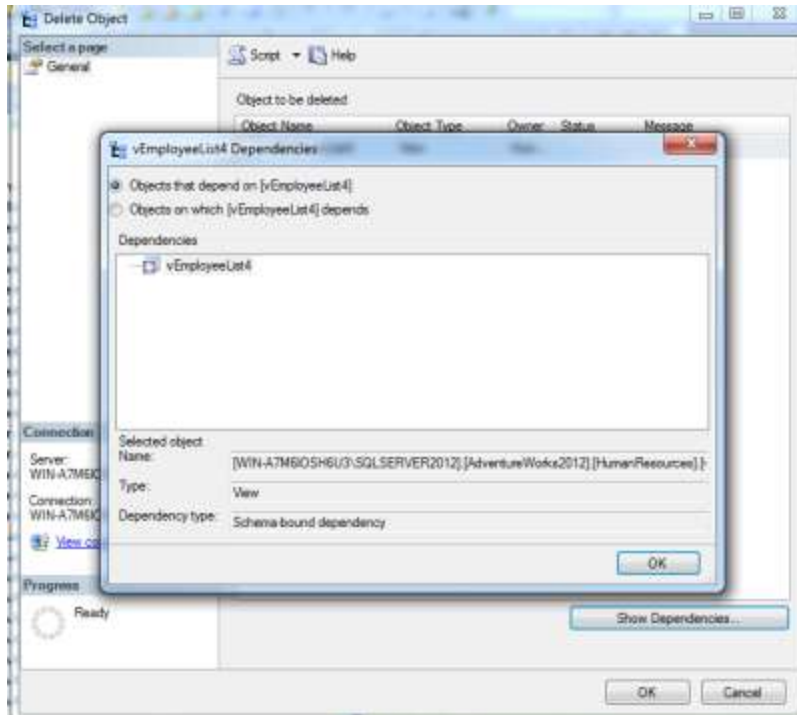
Shouldn't use these for performance and complexity but people often do use these

## Drop Views

Right click on view in object explorer

Can click show dependancies





To do it in T-SQL

**SQL**

```
drop view HumanResources.vEmployeeList2
drop view HumanResources.vEmployeeList3
drop view HumanResources.vEmployeeenane5
```