

Vistas

Una **vista** es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto.

Las vistas permiten:

- **Ocultar información:** permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario opera con los datos de una vista como si se tratara de una tabla, pudiendo modificar tales datos.
- **Simplificar la administración de los permisos de usuario y mejorar la seguridad:** se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.
- **Mejorar el rendimiento:** las consultas se precompilan, con lo que obtenemos un rendimiento mayor.
- **Consolidar el conocimiento de la estructura de la Base de datos:** podemos crear vistas que accedan a los datos básicos de una aplicación, para su uso posterior por otros usuarios o aplicaciones sin necesidad de conocer en profundidad su estructura.

Para crearla...

```
create view NOMBREVISTA as  
SENTENCIASSELECT;
```

Para mostrar los datos a través de la vista...

```
select lista de campos from NOMBREVISTA;
```

Primero probamos una consulta que usamos con frecuencia.

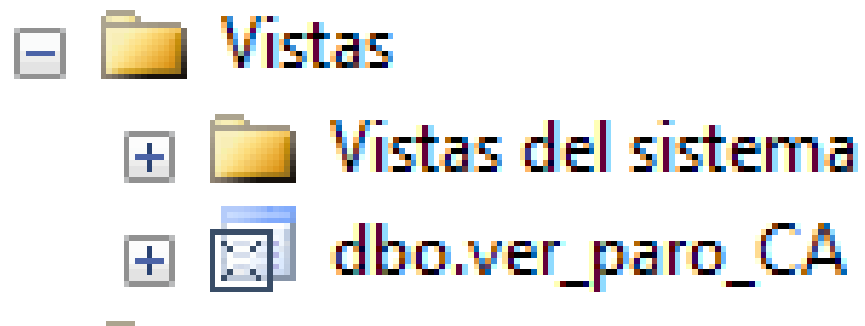
```
use paro  
go
```

```
select ca, sum(padron) as totpadron  
from ComunidadesAutonomas as c  
inner join provincias as p  
    on p.codca=c.codca  
inner join municipios as m  
    on m.codprovincia=p.codprovincia  
inner join padron as pa  
    on pa.codmunicipio=m.codmunicipio  
group by ca;
```

Resultados			Mensajes	
	ca	totpadron		
1	Cantabria	593861		
2	Melilla	80802		
3	Galicia	2781498		
4	Castilla y León	2546078		
5	Madrid, Comunidad de	6498560		
6	Aragón	1349467		
7	Andalucía	8449985		
8	Canarias	2118344		
9	Asturias, Principado de	1077360		
10	País Vasco	2193093		
11	Rioja, La	323609		
12	Extremadura	1110660		

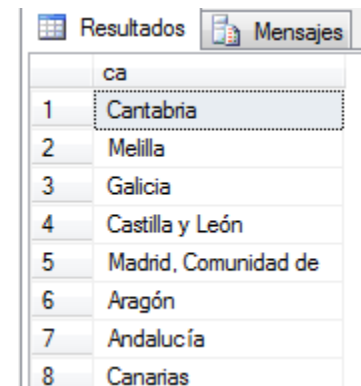
Para convertirla posteriormente en vista...

```
create view ver_paro_CA
as
select ca, sum(padron) as totpadron
from ComunidadesAutonomas as c
inner join provincias as p
    on p.codca=c.codca
inner join municipios as m
    on m.codprovincia=p.codprovincia
inner join padron as pa
    on pa.codmunicipio=m.codmunicipio
group by ca;
go
```



Podemos hacer referencia a ella como si fuera una tabla.

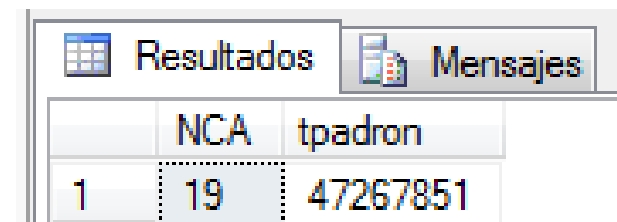
```
select ca from ver_paro_CA;  
go
```



	ca
1	Cantabria
2	Melilla
3	Galicia
4	Castilla y León
5	Madrid, Comunidad de
6	Aragón
7	Andalucía
8	Canarias

Agrupando o calculando...

```
]select count(*) as NCA, sum(totpadron) as tpadron  
-from ver_paro_CA;
```



	NCA	tpadron
1	19	47267851

No podemos asegurar que al añadir una cláusula ORDER BY en la vista se aplique el orden.

Necesitamos poner un TOP para que lo permita.

Valdría un TOP 100 percent, pero en SQL Server no aseguran que se ordene.

Lo mejor es ordenar los resultados y no dentro de la vista.

Se basa en el hecho de que no podemos asegurar el orden de una tabla, por definición de las BD Relacionales y el comportamiento de las vistas es similar.


```

create view ver_paro_CA_o
as
select top 999999999 ca, sum(padron) as totpadron
from ComunidadesAutonomas as c
inner join provincias as p
    on p.codca=c.codca
inner join municipios as m
    on m.codprovincia=p.codprovincia
inner join padron as pa
    on pa.codmunicipio=m.codmunicipio
group by ca
order by totpadron;
go

```

```

select ca, totpadron from ver_paro_CA_o;
go

```

Sale ordenado, con el riesgo de perder algún registro si no entra en el ámbito del top.

Resultados			Mensajes	
	ca	totpadron		
1	Melilla	80802		
2	Ceuta	84018		
3	Rioja, La	323609		
4	Cantabria	593861		
5	Navarra, Comunidad Foral de	644566		
6	Asturias, Principado de	1077360		

```

create view ver_paro_CA_o
as
select top 100 percent ca, sum(padron) as totpadron
from ComunidadesAutonomas as c
inner join provincias as p
    on p.codca=c.codca
inner join municipios as m
    on m.codprovincia=p.codprovincia
inner join padron as pa
    on pa.codmunicipio=m.codmunicipio
group by ca
order by totpadron;
go

```

```

select ca, totpadron from ver_paro_CA_o;
go

```

No da error pero no sale ordenado...

Resultados		Mensajes
	ca	totpadron
1	Cantabria	593861
2	Melilla	80802
3	Galicia	2781498
4	Castilla y León	2546078
5	Madrid, Comunidad de	6498560

Es mejor no ponerle order by a la vista

```
create view ver_paro_CA_o
as
select ca, sum(padron) as totpadron
from ComunidadesAutonomas as c
inner join provincias as p
    on p.codca=c.codca
inner join municipios as m
    on m.codprovincia=p.codprovincia
inner join padron as pa
    on pa.codmunicipio=m.codmunicipio
group by ca;
go
```

```
select ca, totpadron from ver_paro_CA_o
order by totpadron;
go
```

Y ordenar el resultado

Resultados		Mensajes
	ca	totpadron
1	Melilla	80802
2	Ceuta	84018
3	Rioja, La	323609
4	Cantabria	593861
5	Navarra, Comunidad Foral de	644566
6	Asturias, Principado de	1077360

Para borrar una vista haremos.

```
drop view nombre_de_la_vista;
```

```
IF object_id('ver_paro_CA', 'V') IS NOT NULL  
    drop view ver_paro_CA;  
go|
```

Tipo de objeto en object_id:

AF = Función de agregado (CLR)

C = Restricción CHECK

D = DEFAULT (restricción o independiente)

F = Restricción FOREIGN KEY

FN = Función escalar de SQL

FS = Función escalar del ensamblado (CLR)

FT = Función con valores de tabla de ensamblado (CLR)

IF = Función SQL insertada con valores de tabla

IT = Tabla interna

P = Procedimiento almacenado de SQL

PC = Procedimiento almacenado del ensamblado (CLR)

PG = Guía de plan

PK = Restricción PRIMARY KEY

R = Regla (estilo antiguo, independiente)

RF = Procedimiento de filtro de replicación

S = Tabla base del sistema

SN = Sinónimo

SO = Objeto de secuencia

SQ = Cola de servicio

TA = Desencadenador DML del ensamblado (CLR)

TF = Función con valores de tabla SQL

TR = Desencadenador DML de SQL

TT = Tipo de tabla

U = Tabla (definida por el usuario)

UQ = Restricción UNIQUE

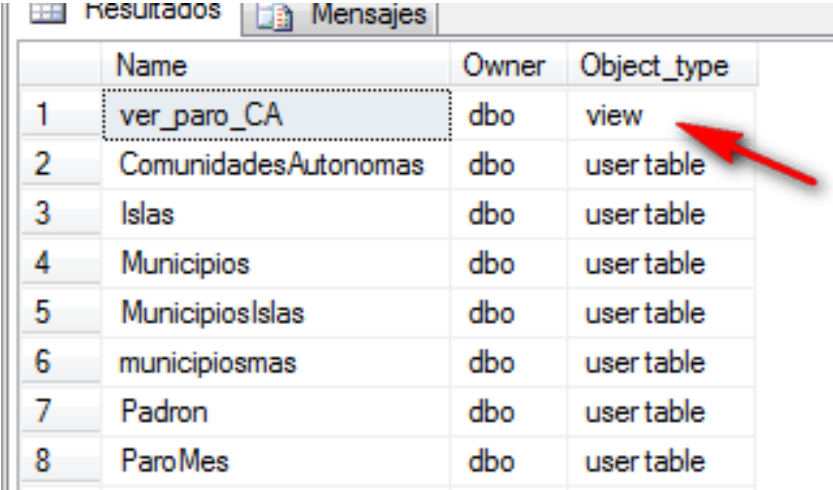
V = Vista

X = Procedimiento almacenado extendido

sp_help

Nos muestra todos los objetos incluidos los de tipo View

```
exec sp_help;  
go
```



	Name	Owner	Object_type
1	ver_paro_CA	dbo	view
2	ComunidadesAutonomas	dbo	user table
3	Islas	dbo	user table
4	Municipios	dbo	user table
5	MunicipiosIslas	dbo	user table
6	municipiosmas	dbo	user table
7	Padron	dbo	user table
8	ParoMes	dbo	user table

sp_helptext nombre_vista

Nos muestra la sentencia que define la vista.

```
exec sp_helptext ver_paro_CA;  
go
```

Resultados		Mensajes
	Text	
1	create view ver_paro_CA	
2	as	
3	select ca, sum(padron) as totpadron	
4	from ComunidadesAutonomas as c	
5	inner join provincias as p	
6	on p.codca=c.codca	
7	inner join municipios as m	
8	on m.codprovincia=p.codprovincia	
9	inner join padron as pa	
10	on pa.codmunicipio=m.codmunicipio	
11	group by ca;	

sp_depends nombre_vista

Nos muestra los campos y objetos de los que depende la vista.

```
exec sp_depends ver_paro_CA;  
go
```

	name	type	updated	selected	column
1	dbo.ComunidadesAutonomas	user table	no	yes	CodCA
2	dbo.ComunidadesAutonomas	user table	no	yes	CA
3	dbo.Municipios	user table	no	yes	CodMunicipio
4	dbo.Municipios	user table	no	yes	CodProvincia
5	dbo.Padron	user table	no	yes	CodMunicipio
6	dbo.Padron	user table	no	yes	Padron
7	dbo.Provincias	user table	no	yes	CodProvincia
8	dbo.Provincias	user table	no	yes	CodCA

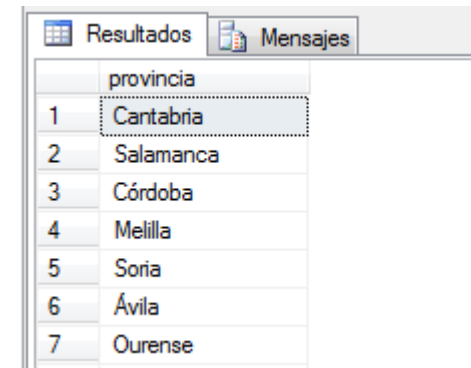
Podemos encriptar la vista para que no se tenga acceso a la sentencia que la define.

```
create view NOMBREVISTA  
with encryption  
as  
SENTENCIASSELECT  
from TABLA;
```

```
create view ver_paro_provincia  
with encryption  
as  
select provincia, sum(padron) as totpadron  
from provincias as p  
inner join municipios as m  
    on m.codprovincia=p.codprovincia  
inner join padron as pa  
    on pa.codmunicipio=m.codmunicipio  
group by provincia;  
go
```

Podemos usarla...

```
select provincia from ver_paro_provincia;  
go|
```

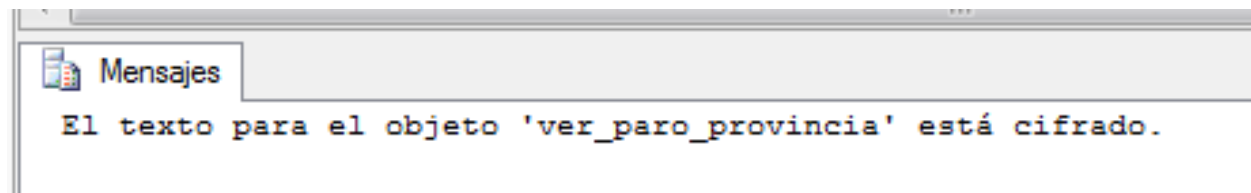


A screenshot of a SQL Server Enterprise Manager window. The 'Resultados' (Results) tab is active, displaying a table with the following data:

	provincia
1	Cantabria
2	Salamanca
3	Córdoba
4	Melilla
5	Soria
6	Ávila
7	Ourense

Pero no podemos ver su sentencia...

```
exec sp_helptext ver_paro_provincia;  
go|
```



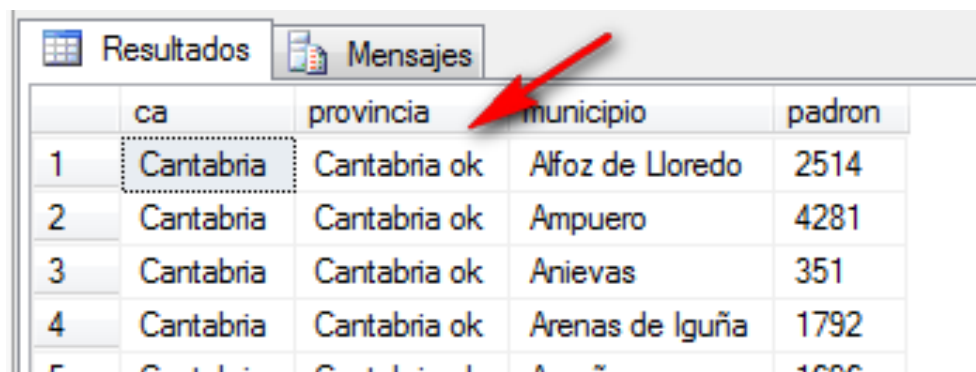
Podemos actualizar datos a través de una vista, tenemos que cuidar que la actualización sólo afecte a una tabla de las consultadas en la vista.

```
create view Ver_datos_padron
as
select ca, provincia,municipio,padron
from ComunidadesAutonomas as c
inner join provincias as p
    on p.codca=c.codca
inner join municipios as m
    on m.codprovincia=p.codprovincia
inner join padron as pa
    on pa.codmunicipio=m.codmunicipio;
go

select ca, provincia,municipio,padron
from Ver_datos_padron;
go
```

```
update ver_datos_padron  
set provincia='Cantabria ok'  
where provincia='Cantabria';  
go
```

```
select ca, provincia,municipio,padron  
from Ver_datos_padron where provincia like 'cant%';  
go
```



	ca	provincia	municipio	padron
1	Cantabria	Cantabria ok	Alfoz de Lloredo	2514
2	Cantabria	Cantabria ok	Ampuero	4281
3	Cantabria	Cantabria ok	Anievas	351
4	Cantabria	Cantabria ok	Arenas de Iguña	1792
5	Cantabria	Cantabria ok	Arenas de Iguña	1000

Se modifica la tabla provincia

No se pueden actualizar vistas con agregados

```
update ver_paro_provincia  
set provincia='Cantabria ok'  
where provincia='Cantabria';  
go
```

Mens. 4403, Nivel 16, Estado 1, Línea 1
No se puede actualizar la vista o función
'ver_paro_provincia' porque contiene
agregados o una cláusula DISTINCT o GROUP
BY, o un operador PIVOT o UNPIVOT.

Para insertar y borrar las vistas deben ser simples.

Hay que poner todos los datos y claves necesarios para que el insert no de error.

```
create view vermunicipio
as
select codmunicipio,municipio,codprovincia
from municipios;
go
```

```
insert into vermunicipio
values (38900,'San Borondón',38);
go|
```

```
delete from vermunicipio
where codmunicipio=38900;
go
```

Si al final de la vista añadimos with check option, sólo podremos hacer operaciones de inserción o actualización que mantengan los datos cumpliendo las condiciones de la vista.

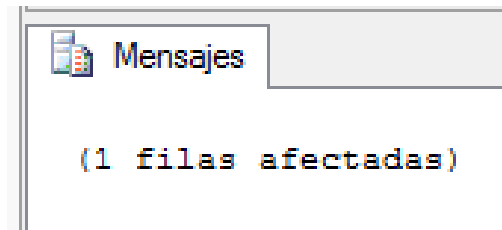
```
--with check option
create view vermunicipio38
as
select codmunicipio,municipio,codprovincia
from municipios
where codprovincia='38'
with check option;
go
```

Los datos cambiados o insertados deberán seguir con codprovincia='38'

```
insert into vermunicipio38  
values (40900, 'San Borondón', 40);  
go
```

Mens. 550, Nivel 16, Estado 1, Línea 1
Error en la inserción o actualización debido a
que la vista de destino especifica WITH CHECK
OPTION o alcanza una vista con esta opción, y
una o más filas resultantes de la operación no
se califican con la restricción CHECK OPTION.

```
insert into vermunicipio38  
values (38902, 'San Borondón city', 38);  
go
```



Podemos modificar una vista borrándola y creándola de nuevo o mediante el alter view.

```
alter view NOMBREVISTA  
with encryption--opcional  
as SELECT
```

```
alter view vermunicipio  
as  
select codmunicipio,municipio,m.codprovincia,provincia  
from municipios as m  
inner join provincias as p  
on p.codprovincia=m.codprovincia;  
go
```