

# Vistas

Una **vista** es una tabla virtual cuyo contenido es el resultado de una consulta. Al igual que una tabla, una vista consta de un conjunto de columnas y filas de datos.

Una vista no existe como conjunto de valores de datos almacenados en una base de datos, sus filas y columnas de datos proceden de las tablas a las que se hace referencia en la consulta que define la vista y se producen de forma dinámica cada vez que se hace referencia a la vista.

La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual o de otras bases de datos.

## Ventajas de las vistas

**Simplicidad:** suelen usarse para centrar, simplificar y personalizar la percepción de la base de datos para cada usuario.

**Seguridad:** Las vistas pueden emplearse como mecanismos de seguridad, permitiendo a los usuarios el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista.

**Rendimiento:** el uso de vistas evita codificar repetidamente una consulta almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

## Sintaxis crear/modificar una vista

Una vista se define usando un **SELECT**.

La sintaxis para **crear o modificar una vista** es la siguiente:

```
create or replace view NOMBRE_VISTA as  
SENTENCIA_SELECT
```

### Ejemplo

```
create or replace view vista_alumnos_casas as  
select a.id, concat_ws(' ',a.nombre,a.apellido1) as  
'alumno',a.idcasa,c.nombre as 'casa'  
from alumno a  
inner join casa c on a.idCasa=c.idcasa;
```

El contenido de una vista se muestra con un **SELECT** igual que hacemos con el contenido de una tabla:

```
Select * from NOMBRE_VISTA
```

#### Ejemplo





```
select *
from vista_alumnos_casas;
```

Los nombres para vistas siguen las mismas reglas que cualquier identificador. Para distinguir una tabla de una vista se suele fijar una convención para darle nombre, por ejemplo, colocar el sufijo “vista” y luego el nombre de las tablas consultadas en ellas. Todos los campos y expresiones de la consulta que define una vista DEBEN tener un nombre, es decir, si tenemos un campo calculado debemos darle un alias para que ese campo en la vista tenga un nombre.

Si tenemos 2 campos con el mismo nombre ,tendremos que utilizar un alias para distinguirlos ya que no podemos tener dos campos con el mismo nombre.

Las vistas se crean sobre la base de datos activa

Al crear una vista, el gestor de BBDD verifica que existan las tablas a las que se hacen referencia en ella, de no ser así se producirá un error (en realidad como cualquier select).

<pre>select a.id, a.idcasa,c.nombre ,a.nombre from alumno a inner join casa c on a.idCasa=c.idcasa;</pre>	
<pre>select a.id, a.idcasa,c.nombre as 'casa' ,a.nombre as 'alumno' from alumno a inner join casa c on a.idCasa=c.idcasa;</pre>	
<pre>select a.id, concat_ws(' ',a.nombre,a.apellido1) ,c.nombre from alumno a inner join casa c on a.idCasa=c.idcasa;</pre>	
<pre>select a.id, concat_ws(' ',a.nombre,a.apellido1) as 'alumno' ,c.nombre as 'casa' from alumno a inner join casa c on a.idCasa=c.idcasa;</pre>	

Otra sintaxis para crear (o modificar) una vista:

```
create or replace view NOMBRE_VISTA (nombreColumna1, nombrecolumna2,...)
  as
  SENTENCIA_SELECT
```

Debemos tener en cuenta que **entre paréntesis tiene que haber tantos nombres como campos o expresiones devuelva la consulta.**

#### Ejemplo

```
create or replace view vista_alumnos_casas
(idalumno,alumno,idcasa,casa) as
select a.id, concat_ws(' ',a.nombre,a.apellido1) ,a.idcasa,c.nombre
from alumno a
inner join casa c on a.idCasa=c.idcasa;
```

#### Sintaxis para eliminar una vista

```
DROP VIEW [if exists] NOMBRE_VISTA;
```

#### Sintaxis para renombrar una vista

```
RENAME TABLE NOMBRE_VISTA TO NUEVO_NOMBRE_VISTA
```