



# TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

**Unidad 7: CREACIÓN Y MANIPULACIÓN DE OTROS  
OBJETOS DE LA BASE DE DATOS: VISTAS,  
USUARIOS, ROLES y PRIVILEGIOS**



# Contenido

1. **Creación de vistas.**
2. **Creación de Usuarios.**
3. **Creación de Roles y Privilegios**

## ¿Qué son las Vistas?

- Una vista es una *ventana* dentro de una tabla.
- Permite aislar el uso de la información protegiéndola y facilitar operaciones que requieran de un modo especial de acceso a las mismas.
- Se forman de la misma manera en que se realizan las selecciones de registros (instrucción **SELECT**) sobre una o mas tablas
- Facilitan el mantenimiento, al poderse modificar las condiciones de selección sin afectar a la estructura de datos retornada.
- Una vista suele llamarse también *tabla virtual*

## 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.
- Simplificar la administración de los permisos de usuario: 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: se puede evitar escribir instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

## Podemos crear vistas con:

- Un **subconjunto** de **registros** y **campos** de una tabla.
- Una **unión** de varias tablas (**UNION** o **UNION ALL**).
- Una **combinación** de varias tablas.
- Un **resumen estadístico** de una tabla (**GROUP BY**).
- Un **subconjunto** de **otra vista**, combinación de vistas y tablas.

Una vista puede tener como máximo 1.024 columnas.

## Sintaxis Genérica:

```
CREATE VIEW [nbEsquema.] nbVista [ (columna [ ,..n ] )  
    ] AS ( sentencia_select ) [ ; ]
```

- ❑ **nbEsquema** : Es el nombre del esquema al que pertenece la nueva tabla.
- ❑ **nbVista** : Es el nombre de la nueva vista. Los nombres de vistas deben seguir las reglas de los identificadores
- ❑ **AS** : Especifica las acciones que va a llevar a cabo la vista.
- ❑ **columna1,.. columnaX** : Encabezados de las columnas que aparecerán en la vista
- ❑ **sentencia\_select** : Es la instrucción SELECT que define la vista.

## Notas:

- ❑ Las **vistas** se crean en la **Base de Datos activa**.
- ❑ Al crear una vista, se **verifica** que **existan** las **tablas** a las que se hacen referencia en ella
  - (Se aconseja probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el imaginado)
- ❑ Existen algunas **restricciones** para el uso de "**create view**":
  - No puede incluir las cláusulas "**compute**" ni "**compute by**" ni la palabra clave "**into**";
  - No se pueden crear **vistas temporales** ni crear vistas sobre tablas temporales.
  - No se pueden asociar **reglas** ni **valores** por defecto a las vistas.
  - No puede combinarse con otras instrucciones en un mismo lote.
- ❑ Se pueden construir **vistas** sobre otras **vistas**.

## Notas:

- ❑ Las **vistas** se crean en la **Base de Datos activa**.
- ❑ Al crear una vista, se **verifica** que **existan** las **tablas** a las que se hacen referencia en ella
  - (Se aconseja probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el imaginado)
- ❑ Existen algunas **restricciones** para el uso de "**create view**":
  - No puede incluir las cláusulas "**compute**" ni "**compute by**" ni la palabra clave "**into**";
  - No se pueden crear **vistas temporales** ni crear vistas sobre tablas temporales.
  - No se pueden asociar **reglas** ni **valores** por defecto a las vistas.
  - No puede combinarse con otras instrucciones en un mismo lote.
- ❑ Se pueden construir **vistas** sobre otras **vistas**.



## Sintaxis Genérica:

```
DROP VIEW [nbEsquema.]nbVista[ ,...n ] [ ; ];
```

- ❑ Se eliminan las vista de la base de datos actual. Cuando eliminamos una vista eliminamos su definición y los permisos asociados a ella.
- ❑ Se pueden quitar varias vistas en una misma sentencia **DROP VIEW** escribiendo los nombres de las vistas a eliminar separados por comas.
- ❑ Para ejecutar **DROP VIEW**, como mínimo, se necesita el permiso ALTER en SCHEMA o el permiso CONTROL en OBJECT.

## USUARIOS: Control de Acceso BD

### Nivel 1: Comprobación de la conexión

- ❑ Desde dónde se conecta el usuario
- ❑ Nombre del usuario
- ❑ Consulta a tabla “user” (host, user, password)

### Nivel 2: Comprobación de privilegios

- ❑ Por cada petición en la conexión se comprueba si hay privilegios para efectuarla
- ❑ Consulta a tablas user, db, tables\_priv, columns\_priv, procs\_priv

# USUARIOS: Control de Acceso BD

## Usuarios del sistema

**SELECT \* from mysql.user;**










✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0.0000 segundos.)

`SELECT * FROM `user` WHERE 1`

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [A](#)

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:  Ordenar según la clave:

+ Opciones

				Host	User	Password	Select_priv
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	localhost	root	*41541B05D70E5AB82E864EA284F2DF6BA02B0611	Y
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	127.0.0.1	root	*41541B05D70E5AB82E864EA284F2DF6BA02B0611	Y
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	:::1	root	*41541B05D70E5AB82E864EA284F2DF6BA02B0611	Y

⬆ ☐ Seleccionar todo Para los elementos que están marcados:  Editar  Copiar  Borrar  Expc

## USUARIOS: Crear Usuario

```
CREATE USER 'nombre_de_usuario'@'host' IDENTIFIED  
BY 'contraseña';
```

```
CREATE USER 'juan'@'localhost' IDENTIFIED BY  
            'manolo';
```

## USUARIOS: Cambiar Contraseña Usuario

Si luego se deseara cambiar (o asignar en caso de inexistencia) una contraseña a un determinado usuario:

```
SET PASSWORD FOR 'usuario'@'host' =  
PASSWORD('nueva clave');
```

```
SET PASSWORD FOR 'juan'@'localhost' =  
PASSWORD('juan');
```

Sólo los usuarios tales como **root** con acceso de modificación para la base de datos mysql puede cambiar la contraseña de otro usuario

## ROLES Y PRIVILEGIOS: CREACION

- ❑ Un nuevo usuario no tiene permisos para hacer algo con las bases de datos.
- ❑ Por consecuencia si el usuario intenta identificarse (con la contraseña establecida), no será capaz de acceder a la consola de MySQL.
- ❑ Por ello, lo primero que debemos hacer es proporcionarle el acceso requerido al usuario con la información que requiere.

```
GRANT [permiso] ON [nombre de bases de  
datos].[nombre de tabla] TO '[nombre de  
usuario]@'localhost' [WITH GRANT OPTION];
```

```
GRANT ALL PRIVILEGES ON * . * TO 'juan'@'localhost';
```

## ROLES Y PRIVILEGIOS: CREACION

### Conceder permiso de conceder permisos

Cuando otorgamos permisos a un determinado usuario, es posible además, conceder un permiso extra que le permita asignar sus mismos privilegios a otros usuarios. Una situación de ejemplo podría ser la siguiente:

Al usuario JUAN le asignamos permisos de selección sobre la base de datos **mediciones**. Entonces, el usuario Ana, puede asignar permisos de selección sobre la base de datos **mediciones** a cualquier otro usuario.

De eso se trata la cláusula **WITH GRANT OPTION** que se especifica de la siguiente manera:

**GRANT** permisos **ON** tablas **TO** usuario **WITH GRANT OPTION**

## ROLES Y PRIVILEGIOS: CREACION

- Una vez que has finalizado con los permisos que deseas configurar para tus nuevos usuarios, hay que asegurarse siempre de refrescar todos los privilegios.

**FLUSH PRIVILEGES;**



## ROLES Y PRIVILEGIOS: CREACION

Aquí está una pequeña lista del resto de los posibles permisos que los usuarios pueden gozar.

- ❑ **ALL PRIVILEGES**: como mencionamos previamente esto permite a un usuario de MySQL acceder a todas las bases de datos asignadas en el sistema.
- ❑ **CREATE**: permite crear nuevas tablas o bases de datos.
- ❑ **DROP**: permite eliminar tablas o bases de datos.
- ❑ **DELETE**: permite eliminar registros de tablas.
- ❑ **INSERT**: permite insertar registros en tablas.
- ❑ **SELECT**: permite leer registros en las tablas.
- ❑ **UPDATE**: permite actualizar registros seleccionados en tablas.
- ❑ **GRANT OPTION**: permite remover privilegios de usuarios.

## ROLES Y PRIVILEGIOS: CREACION

**GRANT all privileges on \*.\* to 'juan'@'localhost'**  
**identified by 'juan';**

**SHOW GRANTS FOR 'juan'@'localhost';**

```
SELECT * from mysql.user
```

## ROLES Y PRIVILEGIOS: REVOCACION

- ❑ Si necesitas remover un permiso, la estructura es casi idéntica a la que los asigna:

```
REVOKE [permiso] ON [nombre de bases de  
datos].[nombre de tabla] FROM '[nombre de  
usuario]@'localhost';
```

```
REVOKE ALL PRIVILEGES ON * . * FROM  
'juan'@'localhost';
```

- ❑ Si necesitas remover un permiso QUE PERMITIA conceder PRIVILEGIOS, la estructura es casi idéntica:

```
REVOKE GRANT OPTION ON * . * FROM  
'juan'@'localhost';
```

## USUARIOS Y ROLES: ELIMINAR

Así como puedes borrar bases de datos con DROP, también puedes usar el comando DROP para borrar usuarios:

```
DROP USER 'nombre_de_usuario'@'host'
```

```
DROP USER 'juan'@'localhost'
```

## NIVELES DE PRIVILEGIOS

### Privilegios Nivel Global

- ❑ Los permisos globales se aplican a todas las bases de datos de un servidor dado.
- ❑ Estos privilegios son almacenados en la tabla “mysql.user”
- ❑ **GRANT ALL ON \*.\* y REVOKE ALL ON \*.\*** otorgan y quitan sólo permisos globales.

**GRANT all privileges on \*.\* to 'juan'@'localhost' with grant option;**

## NIVELES DE PRIVILEGIOS

### Privilegios Nivel de Base de Datos

- ❑ Los permisos de base de datos se aplican a todos los objetos en una base de datos dada
- ❑ Estos permisos se almacenan en las tablas “mysql.db”
- ❑ **GRANT ALL ON db\_name.\*** y **REVOKE ALL ON db\_name.\*** otorgan y quitan sólo permisos de bases de datos.

**GRANT all privileges on consumoelectrico.\* to 'juan'@'localhost' with grant option;**

## NIVELES DE PRIVILEGIOS

### Privilegios Nivel de Tabla

- ❑ Los permisos de tabla se aplican a todas las columnas en una tabla dada
- ❑ Estos permisos se almacenan en la tabla mysql.tables\_priv

```
GRANT ALL ON db_name. tbl_name  
REVOKE ALL ON db_name. tb_name
```

Primeramente:

```
REVOKE ALL PRIVILEGES ON consumoelectrico . * FROM  
'juan'@'localhost';
```

```
GRANT all privileges on consumoelectrico.clientes  
to 'juan'@'localhost' with grant option;
```

## NIVELES DE PRIVILEGIOS

### Privilegios Nivel de Tabla

**GRANT** all privileges on  
consumoelectrico.mediciones **to** 'juan'@'localhost'  
**with grant option**;  
**FLUSH PRIVILEGES**;

Veamos los Privilegios:

**SHOW GRANTS FOR** 'juan'@'localhost';

Revoquemos:

**REVOKE** all privileges consumoelectrico.clientes  
**FROM** 'juan'@'localhost'

Veamos de nuevo los Privilegios:

**SHOW GRANTS FOR** 'juan'@'localhost';



## NIVELES DE PRIVILEGIOS

### Privilegios Nivel de Columnas

- ❑ Los permisos de columna se aplican a columnas en una tabla dada
- ❑ Estos permisos se almacenan en la tabla "mysql.columns\_priv"
- ❑ Usando REVOKE, debe especificar las mismas columnas que se otorgaron los permisos

```
GRANT SELECT (Código, Nombre, Apellido, apellido2) on  
consumoelectrico.Clientes to 'juan'@'localhost';
```

Veamos los Privilegios:

```
SHOW GRANTS FOR 'juan'@'localhost';
```