

# Consulta de creación de tabla

# CREATE TABLE

Puedes crear una **nueva tabla** a partir de una sentencia **SELECT** para poder **duplicar la estructura** completa de la misma o solo parte de ella.

El siguiente ejemplo genera una copia de una tabla existente, incluye **todos sus registros**:

```
CREATE TABLE Clientes_Copia  
SELECT * FROM Clientes;
```



El siguiente ejemplo crea una copia de una tabla existente. Incluye en la nueva tabla solo los registros de la tabla original que satisfacen una condición específica.

```
CREATE TABLE Clientes_Argentina  
SELECT * FROM Clientes WHERE Pais = 'Argentina';
```

**Nota:** en los ejemplos mencionados, se replica la estructura de la tabla original y el tipo de datos de cada uno de sus campos, pero **no la *primary key***.

## Utilización de la IA

Podemos generar un *prompt* en ChatGPT para pedirle que cree una nueva tabla a partir de otra a través de la especificación de algún filtro:

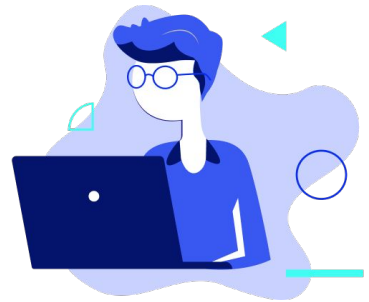
Tengo una tabla de clientes con los [campos id, nombre, apellido, tipo de documento, número de documento, dirección, localidad]. ¿Cómo sería un select en MySQL para obtener otra tabla clientes\_argentina con la misma estructura, pero solo con los clientes cuya localidad sea "Argentina"?

# Consulta de actualización

# Modificar registros de una tabla: *DML UPDATE*

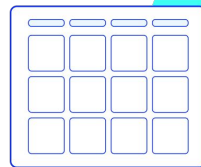
Podemos modificar registros de una tabla a través de la utilización de la sentencia **UPDATE**. La **sintaxis genérica** del comando **UPDATE** para **actualizar un registro** existente es:

```
UPDATE Tabla SET Campo1 = Valor1 WHERE Campo = 'Valor';
```



Es posible, del mismo modo, actualizar el valor de **más de una columna** separándolas en la sección **SET** mediante **comas** como delimitador:

```
UPDATE Tabla SET Campo1 = Valor1, Campo2 = Valor2 WHERE Campo = 'Valor';
```



## Actualizaciones selectivas: cláusula *WHERE*

Mediante la cláusula ***WHERE*** se puede **establecer una condición**; de esta manera, **solo las filas/registros que cumplan esa condición** serán actualizadas.

Ejemplos:

```
UPDATE Articulos SET Nombre = 'Pala Ancha' WHERE Nombre = 'Pala';
```

```
UPDATE Articulos SET Precio = '499.99' WHERE Nombre = 'Pala';
```

```
UPDATE Articulos SET Precio = '499.99' WHERE Nombre <> 'Pala';
```

**Nota:** Si el resultado es **(0 row(s) affected)** quiere decir que **no hay** registros que tengan el valor de la columna indicada en la cláusula ***WHERE***.

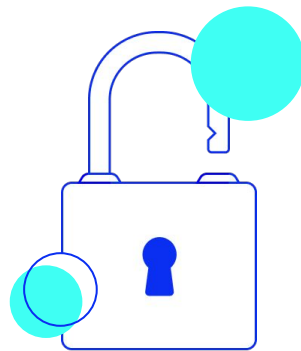


## Desbloqueo de bases de datos

En ocasiones, el motor de base de datos **MySQL Workbench** bloquea **(por defecto) las bases de datos** para evitar actualizaciones o eliminaciones involuntarias.

Por lo tanto, en el caso de querer llevar a cabo la actualización (o eliminación) de registros de una base de datos, será **necesario desbloquear** la base de datos en uso. Para ello, se debe ejecutar la siguiente instrucción:

```
SET SQL_SAFE_UPDATES = 0;
```



# Consulta de eliminación

# Eliminar registros de una tabla: *DML DELETE*

Para eliminar registros de una tabla, se utiliza la sentencia ***DELETE***. Al hacerlo, se deberá especificar la **condición que deben cumplir los registros** de la tabla a eliminar.

En el ejemplo que se muestra a la derecha, se eliminarían de la tabla ***Productos*** todas aquellas filas/registros donde el campo ***idProducto*** contenga el valor **1**.

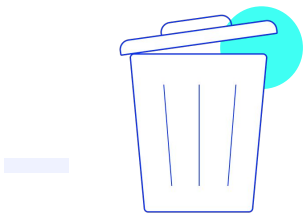
```
DELETE FROM Productos WHERE idProducto = 1;
```

Recuerda que **si no especificas ninguna condición** a través de una cláusula ***WHERE***, **se eliminarán todos los registros** de la tabla sin ninguna limitación (es decir, la tabla quedará vacía).



## TRUNCATE TABLE

La sentencia **DELETE** se puede utilizar para eliminar todos los registros de una tabla, pero tiene la desventaja de no ser tan eficiente, ya que, por ejemplo (entre otras limitaciones), **no resetea los valores de los campos AUTO\_INCREMENT**.



Por este motivo, si se opta por vaciar completamente la tabla, es recomendable utilizar la sentencia **TRUNCATE TABLE**, la cual **elimina los registros en su totalidad y deja vacía la tabla**, de manera menos costosa (en términos de rendimiento) para el servidor de base de datos.

Ejemplo:

```
TRUNCATE TABLE Productos;
```



# Eliminar registros con la IA

En el ámbito de la gestión de bases de datos, **una tarea común es la eliminación de registros específicos o conjuntos de datos que ya no son necesarios o que cumplen ciertas condiciones.**

A continuación, se presentan tres ejemplos de situaciones en las que es necesario eliminar registros en una tabla llamada Clientes.

**Estas consultas permiten desde la eliminación de un único registro, basado en un identificador único, hasta la eliminación masiva según criterios más amplios como fechas o todos los datos de la tabla.**

## ***Prompt 1:***

Hacer una query para una base de datos en MySQL donde se elimine el cliente de la tabla Cliente cuyo identificador sea en número 11,

## ***Prompt 2:***

Hacer una query para una base de datos en MySQL para eliminar todos los clientes de la tabla Clientes cuya fecha de nacimiento sea menor al 01 de Enero del 1980,

## ***Prompt 3:***

Hacer una query para una base de datos en MySQL para eliminar todos los clientes de la tabla Clientes.

**¡Sigamos  
trabajando!**