educación IT
CODE YOUR FUTURE

Bootcamp Java Developer

Fase 1 - Java Analyst

Módulo 5

Consulta de creación de tabla

Copiar una tabla: *CREATE TABLE SELECT*

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

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

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

Y el ejemplo a continuación, genera una copia de una tabla existente, copiando en esta nueva tabla **sólo aquellos registros** de la tabla original que cumplan con una **determinada condición**:

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

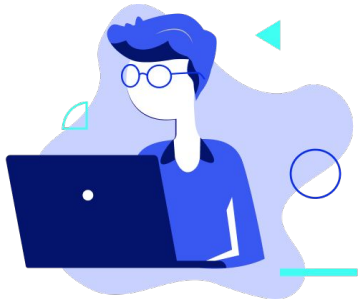
Nota: en los ejemplos mencionados, se replica la estructura de la tabla original y el tipo de datos de cada uno de sus campos, pero **no la *Primary Key***.

Consulta de actualización

Modificar registros de una tabla: *DML UPDATE*

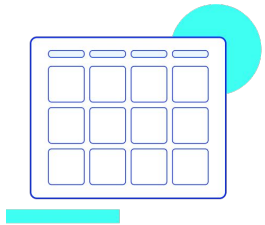
Podemos modificar registros de una tabla utilizando la sentencia ***UPDATE***. La **sintaxis genérica** del comando ***UPDATE*** para **actualizar un registro** existente es la siguiente:

```
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, **sólo las filas/registros que cumplan esa condición** serán actualizadas.

> **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***.

Ejemplos:

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

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

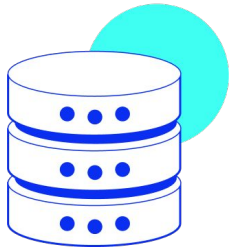
```
UPDATE Articulos SET Precio = '499.99' WHERE Nombre <> 'Pala';
```

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.

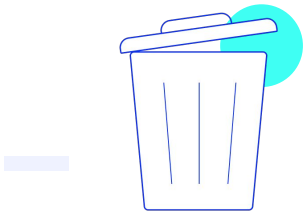
En el ejemplo siguiente, se eliminarían de la tabla ***Productos*** todas aquellas filas/registros donde el campo ***idProducto*** **contenga el valor 1**.

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

Nota: 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:

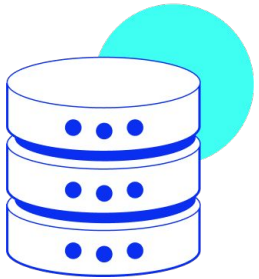
```
TRUNCATE TABLE Productos;
```

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 datos anexados

Consulta de datos anexados: *DML INSERT*

Se pueden **insertar datos** (provenientes de otra tabla) en una tabla a partir de una sentencia ***SELECT***. De este modo, podrás realizar una **inserción masiva de datos desde una tabla hacia otra en una sola instrucción** o sentencia.

Sintaxis:

```
INSERT INTO TablaDestino (Columna1, ..., ColumnaX)
SELECT (Columna1, ..., ColumnaX) FROM TablaOrigen;
```

Nota: la tabla de destino (llamada *TablaDestino* en el ejemplo anterior) debe tener la **misma estructura que la consulta *SELECT***; es decir, la misma cantidad de columnas y tipos de datos compatibles.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!