

CASO PRÁCTICO 1

MODIFICACIONES EN UNA TABLA (ALTER Y UPDATE)

1. Creación de la base de datos

Antes de crear la tabla Productos, es necesario contar con una base de datos donde almacenar la información de la empresa de muebles. Para ello, creo una base de datos llamada **TiendaMuebles** en **MySQL 8.0**.

Comando SQL utilizado

```
mysql> CREATE DATABASE TiendaMuebles;  
Query OK, 1 row affected (0.02 sec)
```

Explicación del comando

- **CREATE DATABASE TiendaMuebles;** crea una nueva base de datos llamada TiendaMuebles.
- No se necesita especificar más detalles, ya que MySQL usa la configuración por defecto para su creación.

Selección de la base de datos

Para trabajar dentro de la base de datos recién creada, utilizo el siguiente comando:

```
mysql> USE TiendaMuebles;  
Database changed
```

Explicación del comando

- **USE TiendaMuebles;** selecciona la base de datos TiendaMuebles para que todos los comandos posteriores (creación de tablas, inserción de datos, etc.) se ejecuten en ella.

Confirmación de la creación

Para verificar que la base de datos se ha creado correctamente, utilizo:

```
mysql> SHOW DATABASES;  
+-----+  
| Database  
+-----+  
| cp_ud5  
| information_schema  
| mysql  
| performance_schema  
| prueba  
| sys  
| tiendamuebles  
+-----+
```

2. Creación de la tabla Productos

Una vez creada la base de datos, procedo a definir la tabla Productos, que contendrá la información de los productos de la tienda de muebles.

Comando SQL utilizado

```
mysql> CREATE TABLE Productos (  
-> idproducto INT PRIMARY KEY AUTO_INCREMENT,  
-> NombreP VARCHAR(100) NOT NULL,  
-> Precio DECIMAL(10,2) NOT NULL,  
-> CategoriaP VARCHAR(50) NOT NULL  
-> );  
Query OK, 0 rows affected (0.04 sec)
```

Explicación del comando

- **CREATE TABLE Productos:** Crea una nueva tabla llamada Productos.
- **idproducto INT AUTO_INCREMENT PRIMARY KEY:** Define la clave primaria, asegurando que cada producto tenga un identificador único y se incremente automáticamente.
- **NombreP VARCHAR(100) NOT NULL:** Almacena el nombre del producto, permitiendo hasta 100 caracteres y evitando valores nulos.
- **Precio DECIMAL(10,2) NOT NULL:** Define el precio con dos decimales para mayor precisión.
- **CategoriaP VARCHAR(50) NOT NULL:** Especifica la categoría del producto con un límite de 50 caracteres.

Confirmación de la creación de la tabla

Tras ejecutar el comando **CREATE TABLE**, MySQL ha confirmado la creación de la tabla Productos sin errores. Para verificar que la estructura es correcta, he utilizado el siguiente comando:

Comando de verificación

```
mysql> DESCRIBE Productos;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| idproducto | int           | NO   | PRI | NULL    | auto_increment |  
| NombreP    | varchar(100)  | NO   |     | NULL    |                |  
| Precio     | decimal(10,2) | NO   |     | NULL    |                |  
| CategoriaP | varchar(50)   | NO   |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

Explicación de la salida

- idproducto está correctamente definido como INT, clave primaria (PRI) y con auto-incremento (AUTO_INCREMENT).
- NombreP, Precio y CategoriaP tienen los tipos de datos correctos y no permiten valores NULL, asegurando la integridad de los datos.

3. Inserción de datos en la tabla

Una vez creada y verificada la tabla Productos, procedo a insertar 5 registros con datos ficticios pero coherentes con el contexto de una tienda de muebles. Cada registro contiene el **nombre del producto**, su **precio** y la **categoría** a la que pertenece.

Comando SQL utilizado

```
mysql> INSERT INTO Productos (NombreP, Precio, CategoriaP) VALUES
-> ('Silla Moderna', 150.00, 'Sillas'),
-> ('Mesa de Madera', 300.00, 'Mesas'),
-> ('Sofá de 3 plazas', 500.00, 'Sofás'),
-> ('Estantería Blanca', 120.00, 'Estanterías'),
-> ('Lámpara LED', 45.00, 'Iluminación');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Explicación del comando

- **INSERT INTO** Productos indica que vamos a insertar nuevos datos en la tabla.
- Se especifican las columnas **NombreP**, **Precio** y **CategoriaP**, ya que **idproducto** se genera automáticamente gracias a AUTO_INCREMENT.
- Se insertan **5 productos** diferentes, con precios realistas y categorías relacionadas con una tienda de muebles.

Confirmación de la inserción

Para confirmar que los datos se han insertado correctamente, utilizo el siguiente comando:

Comando de verificación

```
mysql> SELECT * FROM Productos;
```

idproducto	NombreP	Precio	CategoriaP
1	Silla Moderna	150.00	Sillas
2	Mesa de Madera	300.00	Mesas
3	Sofá de 3 plazas	500.00	Sofás
4	Estantería Blanca	120.00	Estanterías
5	Lámpara LED	45.00	Iluminación

```
5 rows in set (0.00 sec)
```

4. Modificación de la columna (ALTER TABLE)

Tras insertar los datos en la tabla Productos, realizo la modificación solicitada: cambiar el nombre de la columna **idproducto** por **codproducto**.

Comando SQL utilizado

```
mysql> ALTER TABLE Productos CHANGE idproducto codproducto INT;  
Query OK, 5 rows affected (0.12 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

Explicación del comando

- **ALTER TABLE Productos:** Indica que se va a modificar la estructura de la tabla Productos.
- **CHANGE idproducto codproducto INT:**
 - CHANGE se usa para modificar el nombre y el tipo de una columna.
 - Se cambia idproducto por codproducto.
 - Se mantiene el tipo de dato **INT**, asegurando la integridad de los datos.
- Este comando permite a la empresa realizar cambios en la estructura de la base de datos sin perder información, lo que facilita la evolución del sistema según las necesidades del negocio

Confirmación del cambio

Para verificar que la modificación se ha realizado correctamente, utilizo el siguiente comando:

Comando de verificación

```
mysql> DESCRIBE Productos;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| codproducto | int           | NO   | PRI | NULL    |      |  
| NombreP    | varchar(100)  | NO   |     | NULL    |      |  
| Precio     | decimal(10,2) | NO   |     | NULL    |      |  
| CategoriaP | varchar(50)   | NO   |     | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

5. Actualización del precio de un producto (UPDATE)

Como parte de la tarea, aplico un descuento del **20%** al precio de un producto en la tabla Productos. En este caso, el producto seleccionado es la "**Mesa de Madera**".

Comando SQL utilizado

```
mysql> UPDATE Productos  
-> SET Precio = Precio * 0.8  
-> WHERE NombreP = 'Mesa de Madera';  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

Explicación del comando

- **UPDATE Productos:** Indica que se actualizarán datos en la tabla Productos.
- **SET Precio = Precio * 0.8:** Reduce el precio actual en un **20%** (multiplicándolo por 0.8).
- **WHERE NombreP = 'Mesa de Madera':** Asegura que solo se modifique el producto correcto.

Confirmación de la actualización

Para verificar que el precio ha sido actualizado correctamente, utilizo el siguiente comando:

Comando de verificación

```
mysql> SELECT * FROM Productos;
```

codproducto	NombreP	Precio	CategoriaP
1	Silla Moderna	150.00	Sillas
2	Mesa de Madera	240.00	Mesas
3	Sofá de 3 plazas	500.00	Sofás
4	Estantería Blanca	120.00	Estanterías
5	Lámpara LED	45.00	Iluminación

5 rows in set (0.00 sec)

Antes del descuento, el precio de la "Mesa de Madera" era **300.00**. Después de aplicar el **20% de descuento**, el nuevo precio debería ser **240.00**.

Conclusión

En este caso práctico, he llevado a cabo las siguientes operaciones en **MySQL 8.0** utilizando comandos SQL:

- Creación de la base de datos TiendaMuebles.
- Creación de la tabla Productos con sus columnas correctamente definidas.
- Inserción de 5 registros con datos realistas.
- Modificación del nombre de la columna idproducto a codproducto.
- Aplicación de un descuento del **20%** al precio de un producto mediante UPDATE.

Con la verificación de cada paso, confirmo que todas las tareas han sido completadas con éxito.

Referencias y Recursos Consultados

- Documentación oficial de MySQL. (2024). *Manual de referencia de MySQL 8.0*. <https://dev.mysql.com/doc/refman/8.0/en/>
- MySQL Workbench Team. (2024). *Guía del usuario de MySQL Workbench*. <https://dev.mysql.com/doc/workbench/en/>
- W3Schools. (2024). *Tutorial de SQL*. <https://www.w3schools.com/sql/>