

## Procedimientos almacenados:

- Crea un procedimiento almacenado que inserte un nuevo cliente en la tabla Clientes

**Automatic content disabled. Use the manually get the current caret position to toggle automatic content.**

```
1  -- PROCEDIMIENTOS ALMACENADOS
2  -- 1. Insertar nuevo cliente
3  DELIMITER //
4  CREATE PROCEDURE InsertarCliente(
5      IN p_nombre CHAR(100) CHARACTER SET UTF8MB4,
6      IN p_apellido CHAR(100) CHARACTER SET UTF8MB4,
7      IN p_ciudad CHAR(100) CHARACTER SET UTF8MB4
8  )
9  BEGIN
10     INSERT INTO Clientes (Nombre, Apellido, Ciudad)
11     VALUES (p_nombre, p_apellido, p_ciudad);
12 END;
13 //
14 DROP PROCEDURE IF EXISTS InsertarCliente;
15 DELIMITER ;
16
17 -- 2. Actualizar precio de un producto
18 DELIMITER //
19 CREATE PROCEDURE ActualizarPrecioProducto(
20     IN p_productoID INT,
21     IN p_nuevoPrecio DECIMAL(10,2)
22 )
23 BEGIN
24     UPDATE Productos
25     SET Precio = p_nuevoPrecio
26 END;
27 DELIMITER ;
```

**Output**

#	Time	Action
31	11:23:12	CREATE PROCEDURE InsertarCliente( IN p_nombre ...
32	11:23:57	CREATE PROCEDURE InsertarCliente( IN p_nombre ...
33	11:25:22	DROP PROCEDURE IF EXISTS InsertarCliente; CREAT...
34	11:25:44	CREATE PROCEDURE InsertarCliente( IN p_nombre ...

**Herramienta Recortes**

Captura de pantalla copiada en el portapapel  
Guardado automáticamente en la carpeta de capturas de pantalla.

Marcado y uso compartido

```
DELIMITER //
CREATE PROCEDURE InsertarCliente(
    IN p_nombre CHAR(100) CHARACTER SET UTF8MB4,
    IN p_apellido CHAR(100) CHARACTER SET UTF8MB4,
    IN p_ciudad CHAR(100) CHARACTER SET UTF8MB4
)
BEGIN
    INSERT INTO Clientes (Nombre, Apellido, Ciudad)
    VALUES (p_nombre, p_apellido, p_ciudad);
END;
//
DROP PROCEDURE IF EXISTS InsertarCliente;
DELIMITER ;
```

- Crea un procedimiento almacenado que actualice el precio de un producto en la tabla Productos.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'tienda' database selected. The main editor displays a SQL script for 'Query 1'. The script includes comments in Spanish and SQL code to create a stored procedure and a trigger. The 'Output' pane at the bottom shows the execution results of three INSERT statements.

```

16
17  -- 2. Actualizar precio de un producto
18  DELIMITER //
19  CREATE PROCEDURE ActualizarPrecioProducto(
20      IN p_productoID INT,
21      IN p_nuevoPrecio DECIMAL(10,2)
22  )
23  BEGIN
24      UPDATE Productos
25      SET Precio = p_nuevoPrecio
26      WHERE ProductoID = p_productoID;
27  END;
28  //
29  DELIMITER ;
30
31  -- TRIGGERS
32
33  -- 1. Registrar en historial al insertar venta
34  DELIMITER //
35  CREATE TRIGGER trg_InsertarVenta
36  AFTER INSERT ON Ventas
37  FOR EACH ROW
38  BEGIN
39      INSERT INTO HistorialVentas (VentaID, Fecha, Descripcion)
40      VALUES (NEW.VentaID, NEW.Fecha, CONCAT('Venta registrada de prov...
41

```

Output:

#	Time	Action	Message	Duration / Fetch
7	08:43:07	INSERT INTO Clientes (Nombre, Apellido, Ciudad) VALU...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
8	08:43:07	INSERT INTO Productos (Nombre, Precio, Stock) VALU...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
9	08:43:07	INSERT INTO Ventas (ClienteID, ProductoID, Fecha, Ca...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec

```

DELIMITER //
CREATE PROCEDURE ActualizarPrecioProducto(
    IN p_productoID INT,
    IN p_nuevoPrecio DECIMAL(10,2)
)
BEGIN
    UPDATE Productos
    SET Precio = p_nuevoPrecio
    WHERE ProductoID = p_productoID;
END;
//
DELIMITER ;

```

## Triggers:

- Crea un trigger que registre en una tabla HistorialVentas cada vez que se inserte una nueva venta en la tabla Ventas.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'tienda' expanded, showing tables like 'clientes', 'historialventas', 'productos', and 'ventas'. The 'Query 1' editor in the center contains SQL code for creating two triggers. The first trigger, 'trg\_InsertarVenta', is an AFTER INSERT trigger on the 'Ventas' table that inserts data into 'HistorialVentas'. The second trigger, 'trg\_EvitarEliminarProducto', is a BEFORE DELETE trigger on the 'Productos' table. The 'Output' pane at the bottom shows a log of database actions, including the successful creation of the first trigger.

```
31
32  -- TRIGGERS
33
34  -- 1. Registrar en historial al insertar venta
35  DELIMITER //
36  CREATE TRIGGER trg_InsertarVenta
37  AFTER INSERT ON Ventas
38  FOR EACH ROW
39  BEGIN
40      INSERT INTO HistorialVentas (VentaID, Fecha, Descripcion)
41      VALUES (NEW.VentaID, NEW.Fecha, CONCAT('Venta registrada de prod
42  END;
43  //
44  DELIMITER ;
45
46  -- 2. Impedir eliminación de producto si stock < 10
47
48  DELIMITER //
49  CREATE TRIGGER trg_EvitarEliminarProducto
50  BEFORE DELETE ON Productos
```

Column: Description  
Collation: utf8mb3\_general\_ci  
Definition: Description varchar(255)

#	Time	Action	Message	Duration / Fetch
11	09:52:29	CREATE PROCEDURE ActualizarPrecioProducto( IN ...	0 row(s) affected	0.016 sec
12	09:52:29	CREATE TRIGGER trg_InsertarVenta AFTER INSERT O...	0 row(s) affected	0.016 sec
13	09:52:29	CREATE TRIGGER trg_EvitarEliminarProducto BEFORE ...	0 row(s) affected	0.016 sec
14	09:52:29	CREATE FUNCTION TotalVentasCliente(p_clienteID INT...	0 row(s) affected	0.015 sec
15	09:52:29	CREATE FUNCTION NombreCompletoCliente(p_clienteID INT...	0 row(s) affected, 2 warning(s): 3720 NATIONAL/NCHA...	0.016 sec
16	09:52:29	-- VISTAS -- 1. Productos con precio superior a 50€ CRE...	0 row(s) affected	0.015 sec
17	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Ciudades ...	0 row(s) affected	0.000 sec
18	09:52:29	CREATE OR REPLACE VIEW Vista_Top5_Productos A...	0 row(s) affected	0.016 sec
19	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Fieles AS ...	0 row(s) affected	0.015 sec

```
DELIMITER //
CREATE TRIGGER tr_InsertarVenta
AFTER INSERT ON Ventas
FOR EACH ROW
BEGIN
    INSERT INTO HistorialVentas (VentaID, Fecha, Descripcion)
    VALUES (NEW.VentaID, NEW.Fecha, CONCAT('Venta registrada de producto ',
NEW.ProductoID));
END;
//
DELIMITER ;
```

- Crea un trigger que impida la eliminación de productos con un stock inferior a 10 unidades.

The screenshot shows the MySQL Workbench interface. The Navigator on the left displays the database schema, including tables like 'clientes', 'productos', and 'ventas'. The SQL Editor in the center contains the following SQL code:

```

46 -- 2. Impedir eliminación de producto si stock < 10
47
48 DELIMITER //
49 CREATE TRIGGER trg_EvitarEliminarProducto
50 BEFORE DELETE ON Productos
51 FOR EACH ROW
52 BEGIN
53     IF OLD.Stock < 10 THEN
54         SIGNAL SQLSTATE '45000'
55         SET MESSAGE_TEXT = 'No se puede eliminar un producto con stock inferior a 10 unidades.';
56     END IF;
57 END;
58 //
59 DELIMITER ;
60
61 -- FUNCIONES
62
63
64 -- 1. Calcular total de ventas de un cliente
65 DELIMITER //

```

The Output window at the bottom shows the execution results of the SQL script. It lists 19 actions, including the creation of the trigger 'trg\_EvitarEliminarProducto' at line 49. The output table has columns: #, Time, Action, Message, and Duration / Fetch.

#	Time	Action	Message	Duration / Fetch
11	09:52:29	CREATE PROCEDURE ActualizarPrecioProducto( IN ...	0 row(s) affected	0.016 sec
12	09:52:29	CREATE TRIGGER trg_InserirVenta AFTER INSERT O...	0 row(s) affected	0.016 sec
13	09:52:29	CREATE TRIGGER trg_EvitarEliminarProducto BEFORE ...	0 row(s) affected	0.016 sec
14	09:52:29	CREATE FUNCTION TotalVentasCliente(p_clienteID INT...	0 row(s) affected	0.015 sec
15	09:52:29	CREATE FUNCTION NombreCompletoCliente(p_clienteID ...	0 row(s) affected, 2 warning(s): 3720 NATIONAL/NCHA...	0.016 sec
16	09:52:29	-- VISTAS -- 1. Productos con precio superior a 50€ CRE...	0 row(s) affected	0.015 sec
17	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Ciudades ...	0 row(s) affected	0.000 sec
18	09:52:29	CREATE OR REPLACE VIEW Vista_Top5_Productos A...	0 row(s) affected	0.016 sec
19	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Fieles AS ...	0 row(s) affected	0.015 sec

```

DELIMITER //
CREATE TRIGGER tr_EvitarEliminarProducto
BEFORE DELETE ON Productos
FOR EACH ROW
BEGIN
    IF OLD.Stock < 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No se puede eliminar un producto con stock inferior a 10
unidades.';
    END IF;
END;
//
DELIMITER ;

```



## Funciones:

- Crea una función que calcule el total de ventas en un cliente dado su ID.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with the 'tienda' database selected. The main editor window shows a SQL script for creating a function. The script is as follows:

```
61
62 -- FUNCIONES
63
64 -- 1. Calcular total de ventas de un cliente
65 DELIMITER //
66 CREATE FUNCTION TotalVentasCliente(p_clienteID INT)
67 RETURNS DECIMAL(10,2)
68 DETERMINISTIC
69 BEGIN
70     DECLARE total DECIMAL(10,2);
71     SELECT SUM(V.Cantidad * P.Precio)
72     INTO total
73     FROM Ventas V
74     JOIN Productos P ON V.ProductoID = P.ProductoID
75     WHERE V.ClienteID = p_clienteID;
76     RETURN IFNULL(total, 0);
77 END;
78 //
79 DELIMITER ;
80
```

The right sidebar shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

The bottom panel shows the 'Output' tab with a table of actions and their results:

Action	Time	Message	Duration / Fetch
CREATE PROCEDURE ActualizarPrecioProducto( IN ...	09:52:29	0 row(s) affected	0.016 sec
CREATE TRIGGER trig_InsertarVenta AFTER INSERT O...	09:52:29	0 row(s) affected	0.016 sec
CREATE TRIGGER trig_EvitarEliminarProducto BEFORE ...	09:52:29	0 row(s) affected	0.016 sec
CREATE FUNCTION TotalVentasCliente(p_clienteID INT...	09:52:29	0 row(s) affected	0.015 sec
CREATE FUNCTION NombreCompletoCliente(p_clienteID...	09:52:29	0 row(s) affected, 2 warning(s): 3720 NATIONAL/NCHA...	0.016 sec
-- VISTAS -- 1. Productos con precio superior a 50€ CRE...	09:52:29	0 row(s) affected	0.015 sec
CREATE OR REPLACE VIEW Vista_Clientes_Ciudades ...	09:52:29	0 row(s) affected	0.000 sec
CREATE OR REPLACE VIEW Vista_Top5_Productos A...	09:52:29	0 row(s) affected	0.016 sec
CREATE OR REPLACE VIEW Vista_Clientes_Fieles AS ...	09:52:29	0 row(s) affected	0.015 sec

```
DELIMITER //
CREATE FUNCTION TotalVentasCliente(p_clienteID INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    DECLARE total DECIMAL(10,2);
    SELECT SUM(V.Cantidad * P.Precio)
    INTO total
    FROM Ventas V
    JOIN Productos P ON V.ProductoID = P.ProductoID
    WHERE V.ClienteID = p_clienteID;
    RETURN IFNULL(total, 0);
END;
//
DELIMITER ;
```

- Crear una función que devuelva el nombre completo de un cliente dado su ID.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'tienda' database selected. The main editor shows a SQL query with the following code:

```

75 WHERE V.ClienteID = p_clienteID;
76 RETURN IFNULL(total, 0);
77 END;
78 //
79 DELIMITER ;
80 -- 2. Obtener nombre completo de cliente
81 DELIMITER //
82 CREATE FUNCTION NombreCompletoCliente(p_clienteID INT)
83 RETURNS CHAR(200) CHARACTER SET utf8mb4
84 DETERMINISTIC
85 BEGIN
86 DECLARE nombreCompleto CHAR(200) CHARACTER SET utf8mb4;
87 SELECT CONCAT(Nombre, ' ', Apellido)
88 INTO nombreCompleto
89 FROM Clientes
90 WHERE ClienteID = p_clienteID;
91 RETURN nombreCompleto;
92 END;

```

On the right, a context help window is visible with the text: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

At the bottom, the 'Output' pane shows the execution results of the query:

#	Time	Action	Message	Duration / Fetch
17	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Ciudades ...	0 row(s) affected	0.000 sec
18	09:52:29	CREATE OR REPLACE VIEW Vista_Top5_Productos A...	0 row(s) affected	0.016 sec
19	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Fieles AS ...	0 row(s) affected	0.015 sec
20	10:58:15	CREATE PROCEDURE InsertarCliente( IN p_nombre ...	Error Code: 1304. PROCEDURE InsertarCliente already e...	0.125 sec
21	10:59:18	CREATE PROCEDURE InsertarCliente( IN p_nombre ...	Error Code: 1304. PROCEDURE InsertarCliente already e...	0.000 sec
22	11:00:08	DROP PROCEDURE IF EXISTS InsertarCliente	0 row(s) affected	0.031 sec
23	11:00:48	CREATE PROCEDURE InsertarCliente( IN p_nombre ...	0 row(s) affected, 3 warning(s): 3720 NATIONAL/NCHA...	0.078 sec
24	11:04:53	CREATE FUNCTION NombreCompletoCliente(p_clienteI...	Error Code: 1304. FUNCTION NombreCompletoCliente al...	0.000 sec
25	11:11:26	DROP FUNCTION IF EXISTS NombreCompletoCliente	0 row(s) affected	0.016 sec
26	11:11:43	CREATE FUNCTION NombreCompletoCliente(p_clienteI...	0 row(s) affected	0.046 sec

DELIMITER //

CREATE FUNCTION NombreCompletoCliente(p\_clienteID INT)

RETURNS CHAR(200) CHARACTER SET utf8mb4

DETERMINISTIC

BEGIN

DECLARE nombreCompleto CHAR(200) CHARACTER SET utf8mb4;

SELECT CONCAT(Nombre, ' ', Apellido)

INTO nombreCompleto

FROM Clientes

WHERE ClienteID = p\_clienteID;

RETURN nombreCompleto;

END;

DELIMITER ;

## Vistas:

- Crea una vista que muestre los productos con un precio superior a 501€.
- Crea una vista que muestre los clientes y sus respectivas ciudades.
- Crea una vista que devuelva los 5 productos mas vendidos.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'tienda' database selected. The central 'Query 1' window contains SQL code for creating three views. The right sidebar shows a message about automatic context help being disabled. The bottom 'Output' window displays the execution results of the SQL statements.

```
-- VISTAS

-- 1. Productos con precio superior a 50€
CREATE OR REPLACE VIEW Vista_Productos_Caros AS
SELECT * FROM Productos
WHERE Precio > 50;

-- 2. Clientes con sus ciudades
CREATE OR REPLACE VIEW Vista_Clientes_Ciudades AS
SELECT Nombre, Apellido, Ciudad FROM Clientes;

-- 3. Los 5 productos más vendidos
CREATE OR REPLACE VIEW Vista_Top5_Productos AS
SELECT P.ProductoID, P.Nombre, SUM(V.Cantidad) AS TotalVendidas
FROM Ventas V
JOIN Productos P ON V.ProductoID = P.ProductoID
GROUP BY P.ProductoID, P.Nombre
ORDER BY TotalVendidas DESC
LIMIT 5;
```

#	Time	Action	Message	Duration / Fetch
11	09:52:29	CREATE PROCEDURE ActualizarPrecioProducto( IN ...	0 row(s) affected	0.016 sec
12	09:52:29	CREATE TRIGGER trg_insertarVenta AFTER INSERT O...	0 row(s) affected	0.016 sec
13	09:52:29	CREATE TRIGGER trg_EvitarEliminarProducto BEFORE ...	0 row(s) affected	0.016 sec
14	09:52:29	CREATE FUNCTION TotalVentasCliente(p_clienteID INT...	0 row(s) affected	0.015 sec
15	09:52:29	CREATE FUNCTION NombreCompletoCliente(p_clienteID INT...	0 row(s) affected, 2 warning(s): 3720 NATIONAL/NCHA...	0.016 sec
16	09:52:29	-- VISTAS -- 1. Productos con precio superior a 50€ CRE...	0 row(s) affected	0.015 sec
17	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Ciudades ...	0 row(s) affected	0.000 sec
18	09:52:29	CREATE OR REPLACE VIEW Vista_Top5_Productos A...	0 row(s) affected	0.016 sec
19	09:52:29	CREATE OR REPLACE VIEW Vista_Clientes_Files AS ...	0 row(s) affected	0.015 sec

-- VISTAS

-- 1. Productos con precio superior a 50€

```
CREATE OR REPLACE VIEW Vista_Productos_Caros AS
SELECT * FROM Productos
```

```
WHERE Precio > 50;
```

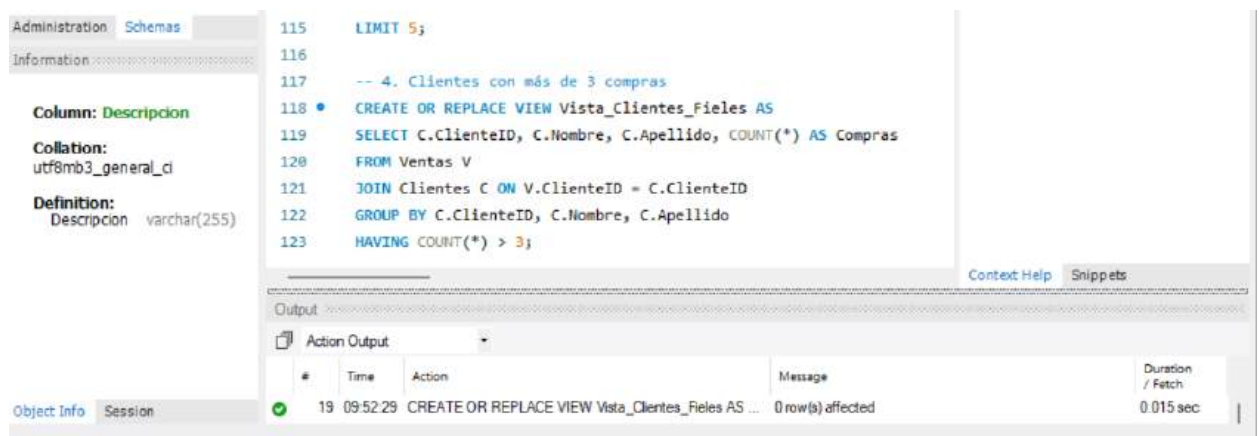
```
-- 2. Clientes con sus ciudades
```

```
CREATE OR REPLACE VIEW Vista_Clientes_Ciudades AS  
SELECT Nombre, Apellido, Ciudad FROM Clientes;
```

```
-- 3. Los 5 productos más vendidos
```

```
CREATE OR REPLACE VIEW Vista_Top5_Productos AS  
SELECT P.ProductoID, P.Nombre, SUM(V.Cantidad) AS TotalVendidas  
FROM Ventas V  
JOIN Productos P ON V.ProductoID = P.ProductoID  
GROUP BY P.ProductoID, P.Nombre  
ORDER BY TotalVendidas DESC  
LIMIT 5;
```

- Crea una vista que devuelva los clientes que han realizado mas de 3 compras.



```
CREATE OR REPLACE VIEW Vista_Clientes_Fieles AS  
SELECT C.ClienteID, C.Nombre, C.Apellido, COUNT(*) AS Compras  
FROM Ventas V  
JOIN Clientes C ON V.ClienteID = C.ClienteID  
GROUP BY C.ClienteID, C.Nombre, C.Apellido  
HAVING COUNT(*) > 3;
```