

EJEMPLO DE PROCEDIMIENTO ALMACENADO

```
DROP DATABASE IF EXISTS ventasprogramacion;
```

```
CREATE DATABASE ventasprogramacion CHARACTER SET utf8mb4;
```

```
USE ventasprogramacion;
```

```
CREATE TABLE cliente (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido1 VARCHAR(100) NOT NULL,  
    apellido2 VARCHAR(100),  
    ciudad VARCHAR(100),  
    categoria INT UNSIGNED  
);
```

```
CREATE TABLE comercial (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido1 VARCHAR(100) NOT NULL,  
    apellido2 VARCHAR(100),  
    comision FLOAT  
);
```

```
CREATE TABLE pedido (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    total DOUBLE NOT NULL,  
    fecha DATE,  
    id_cliente INT UNSIGNED NOT NULL,  
    id_comercial INT UNSIGNED NOT NULL,  
    FOREIGN KEY (id_cliente) REFERENCES cliente(id),  
    FOREIGN KEY (id_comercial) REFERENCES comercial(id)  
);
```

```

INSERT INTO cliente VALUES(1, 'Aarón', 'Rivero', 'Gómez', 'Almería', 100);
INSERT INTO cliente VALUES(2, 'Adela', 'Salas', 'Díaz', 'Granada', 200);
INSERT INTO cliente VALUES(3, 'Adolfo', 'Rubio', 'Flores', 'Sevilla', NULL);
INSERT INTO cliente VALUES(4, 'Adrián', 'Suárez', NULL, 'Jaén', 300);
INSERT INTO cliente VALUES(5, 'Marcos', 'Loyola', 'Méndez', 'Almería', 200);
INSERT INTO cliente VALUES(6, 'María', 'Santana', 'Moreno', 'Cádiz', 100);
INSERT INTO cliente VALUES(7, 'Pilar', 'Ruiz', NULL, 'Sevilla', 300);
INSERT INTO cliente VALUES(8, 'Pepe', 'Ruiz', 'Santana', 'Huelva', 200);
INSERT INTO cliente VALUES(9, 'Guillermo', 'López', 'Gómez', 'Granada', 225);
INSERT INTO cliente VALUES(10, 'Daniel', 'Santana', 'Loyola', 'Sevilla', 125);
INSERT INTO comercial VALUES(1, 'Daniel', 'Sáez', 'Vega', 0.15);
INSERT INTO comercial VALUES(2, 'Juan', 'Gómez', 'López', 0.13);
INSERT INTO comercial VALUES(3, 'Diego', 'Flores', 'Salas', 0.11);
INSERT INTO comercial VALUES(4, 'Marta', 'Herrera', 'Gil', 0.14);
INSERT INTO comercial VALUES(5, 'Antonio', 'Carretero', 'Ortega', 0.12);
INSERT INTO comercial VALUES(6, 'Manuel', 'Domínguez', 'Hernández', 0.13);
INSERT INTO comercial VALUES(7, 'Antonio', 'Vega', 'Hernández', 0.11);
INSERT INTO comercial VALUES(8, 'Alfredo', 'Ruiz', 'Flores', 0.05);
INSERT INTO pedido VALUES(1, 150.5, '2017-10-05', 5, 2);
INSERT INTO pedido VALUES(2, 270.65, '2016-09-10', 1, 5);
INSERT INTO pedido VALUES(3, 65.26, '2017-10-05', 2, 1);
INSERT INTO pedido VALUES(4, 110.5, '2016-08-17', 8, 3);
INSERT INTO pedido VALUES(5, 948.5, '2017-09-10', 5, 2);
INSERT INTO pedido VALUES(6, 2400.6, '2016-07-27', 7, 1);
INSERT INTO pedido VALUES(7, 5760, '2015-09-10', 2, 1);
INSERT INTO pedido VALUES(8, 1983.43, '2017-10-10', 4, 6);
INSERT INTO pedido VALUES(9, 2480.4, '2016-10-10', 8, 3);
INSERT INTO pedido VALUES(10, 250.45, '2015-06-27', 8, 2);
INSERT INTO pedido VALUES(11, 75.29, '2016-08-17', 3, 7);
INSERT INTO pedido VALUES(12, 3045.6, '2017-04-25', 2, 1);

```

```
INSERT INTO pedido VALUES(13, 545.75, '2019-01-25', 6, 1);
INSERT INTO pedido VALUES(14, 145.82, '2017-02-02', 6, 1);
INSERT INTO pedido VALUES(15, 370.85, '2019-03-11', 1, 5);
INSERT INTO pedido VALUES(16, 2389.23, '2019-03-11', 1, 5);
```

```
COMMIT;
```

/*-- 1. Procedimiento "mostrar_clientes" que muestre los nombres y apellidos de clientes ordenados por sus apellidos.*/

```
DROP PROCEDURE IF EXISTS mostrar_clientes;
DELIMITER $$
```

```
CREATE PROCEDURE mostrar_clientes()
BEGIN
    SELECT nombre, apellido1, apellido2
    FROM cliente
    ORDER BY apellido1, apellido2;
END$$
DELIMITER ;
```

/* LLAMADAS AL PROCEDIMIENTO ALMACENADO*/

```
CALL mostrar_clientes();
```

/* Explicación

En este código:

- Primero, cambie el delimitador predeterminado a \$\$.
- En segundo lugar, utilice (;) en el cuerpo del procedimiento almacenado y \$\$ después de la END palabra clave para finalizar el procedimiento almacenado.
- En tercer lugar, vuelva a cambiar el delimitador predeterminado a un punto y coma (;)

***/**

/*-- 2. Procedimiento "mostrar_clientes_ciudad" que, dada una ciudad, muestre los nombres y apellidos de clientes ordenados por sus apellidos

-- Ejemplo: CALL mostrar_clientes_ciudad('SEVILLA');*/

DROP PROCEDURE IF EXISTS mostrar_clientes_ciudad;

DELIMITER \$\$

CREATE PROCEDURE mostrar_clientes_ciudad (ciudad VARCHAR(100))

COMMENT 'LISTA DE LOS CLIENTES CUYA CIUDAD SEA'

BEGIN

SELECT c.nombre , c.apellido1 , c.apellido2

FROM cliente c

WHERE c.ciudad = ciudad

ORDER BY apellido1,apellido2;

END\$\$

DELIMITER ;

CALL mostrar_clientes_ciudad ('SEVILLA');

/*-- 3. Procedimiento que muestra los comerciales que tendrán una comisiones mínimas se le ha de pasar una comisión como parámetro */

-- Ejemplo: CALL mostrar_comerciales_comision_minima (0.12);*/

/*-- 4. Procedimiento "mostrar_pedidos" que, dado un año de inicio y final, muestre los pedidos ordenados por fechas

-- Ejemplo: CALL mostrar_pedidos(2015,2017);*/

/*-- 5. Procedimiento que, dada una ciudad, muestre los nombres y apellidos de clientes ordenados por sus apellidos, en el caso de que no exista ningún cliente de la ciudad deberá mostrar: "NO HAY CLIENTES DE LA CIUDAD <nombreciudad>"

-- Ejemplo: CALL mostrar_clientes_ciudad('SEVILLA');*/

/*-- 6. Procedimiento "mostrar_numpedidos_fechas", que, dado una fecha inicial y otra final, muestre los pedidos ordenados por fechas y si no tuviera pedidos mostrar: "NO HAY PEDIDOS REALIZADOS".

-- El procedimiento deberá comprobar si la fecha de inicio es menor que la fecha final y mostrar un mensaje si no se cumple.

CALL mostrar_numpedidos_fechas('2018-01-01', '2020-01-01');*/