**Propotipo Proyecto Final**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**José Domingo Olea Sarmiento**

**Juan Eduardo Jattin Ceballos**

**Bayron Paternina Morales**

**Rubén Enrique Baena Navarro**

**Base de datos II**

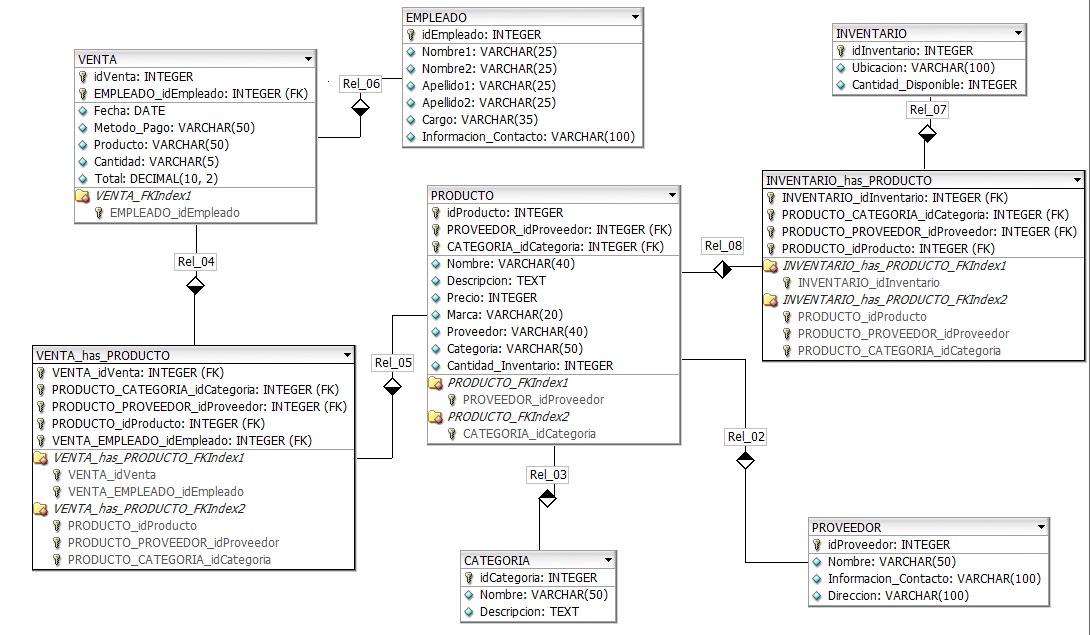
**Universidad de Córdoba**

**Facultad de ingenierías**

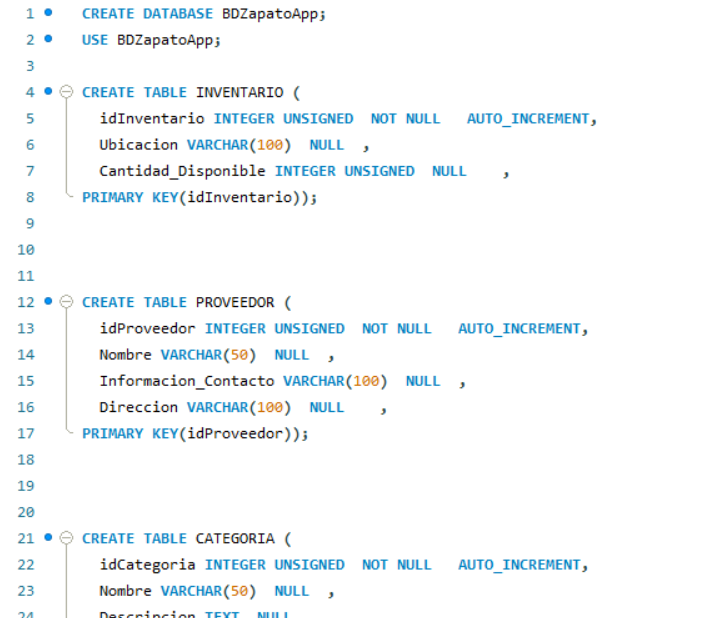
**Programa ingeniería de sistemas**

**2023**

**Modelo Entidad-Relación**



Base de datos creada:



**Tablas:**

**categorías:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**empleados:**

Tabla

Descripción generada automáticamente con confianza media

**inventario:**

**Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente**

**productos:**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Proveedores:**

**Texto

Descripción generada automáticamente**

**Ventas:**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**SUBCONSULTAS:**

* **Para obtener los tres zapatos más vendidos**

SELECT Nombre, Cantidad\_Vendida

FROM (

SELECT Producto, COUNT(\*) AS Cantidad\_Vendida

FROM VENTA

GROUP BY Producto

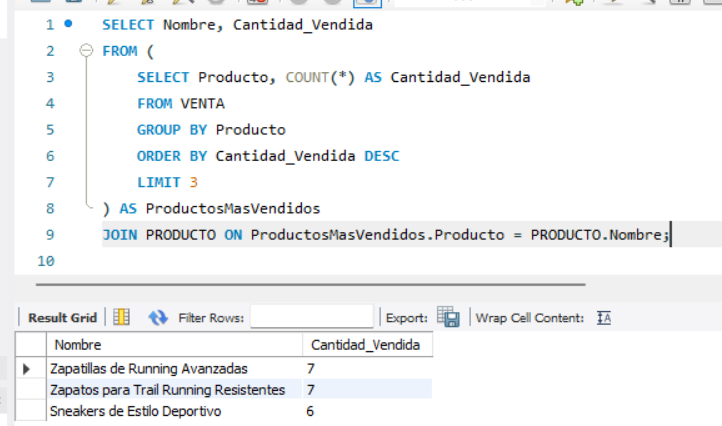
ORDER BY Cantidad\_Vendida DESC

LIMIT 3

) AS ProductosMasVendidos

JOIN PRODUCTO ON ProductosMasVendidos.Producto = PRODUCTO.Nombre;

**Resultado:**



* **Obtener el nombre de los empleados que realizaron ventas y su total de ventas:**

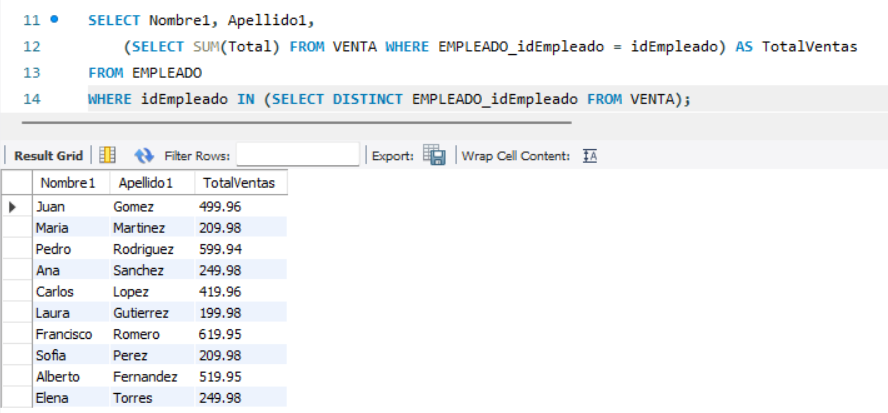
SELECT Nombre1, Apellido1,

(SELECT SUM(Total) FROM VENTA WHERE EMPLEADO\_idEmpleado = idEmpleado) AS TotalVentas

FROM EMPLEADO

WHERE idEmpleado IN (SELECT DISTINCT EMPLEADO\_idEmpleado FROM VENTA);

**Resultado:**



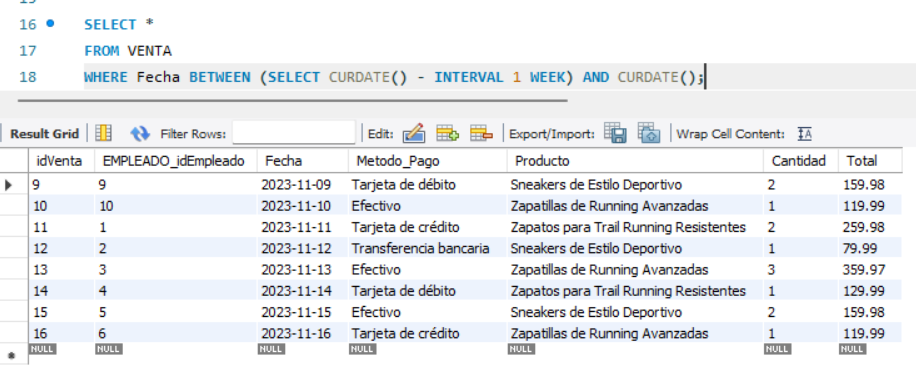
* **Obtener las ventas de la última semana**

SELECT \*

FROM VENTA

WHERE Fecha BETWEEN (SELECT CURDATE() - INTERVAL 1 WEEK) AND CURDATE();

**Resultado:**



* **Para obtener los 3 zapatos menos vendidos**

SELECT Nombre, Cantidad\_Vendida

FROM (

SELECT Producto, COUNT(\*) AS Cantidad\_Vendida

FROM VENTA

GROUP BY Producto

ORDER BY Cantidad\_Vendida ASC

LIMIT 1

) AS ProductosMenosVendidos

JOIN PRODUCTO ON ProductosMenosVendidos.Producto = PRODUCTO.Nombre;

**Resultado:**



**PROCEDIMIENTOS ALMACENADOS:**

* **Procedimiento para insertar una nueva venta**

DELIMITER //

CREATE PROCEDURE InsertarVenta(

IN empleadoID INT,

IN fechaVenta DATE,

IN metodoPago VARCHAR(50),

IN producto VARCHAR(50),

IN cantidad VARCHAR(5),

IN total DECIMAL(10, 2)

)

BEGIN

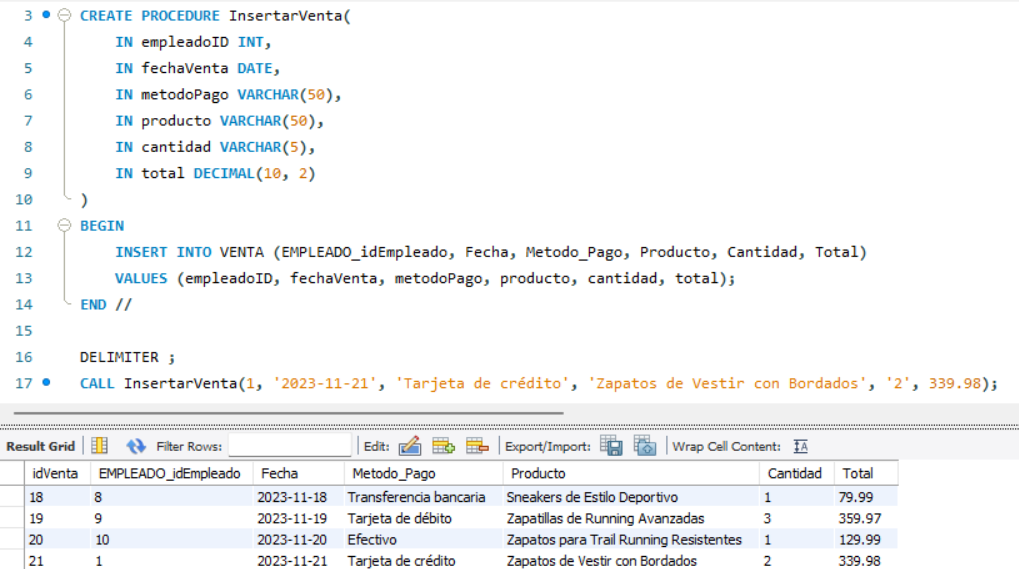
INSERT INTO VENTA (EMPLEADO\_idEmpleado, Fecha, Metodo\_Pago, Producto, Cantidad, Total)

VALUES (empleadoID, fechaVenta, metodoPago, producto, cantidad, total);

END //

DELIMITER ;

**Resultado:**

****

* **Procedimiento para actualizar la cantidad disponible en el inventario después de una venta**

DELIMITER //

CREATE PROCEDURE ActualizarInventarioDespuesVenta(

IN productoID INT,

IN cantidadVendida INT

)

BEGIN

UPDATE INVENTARIO

SET Cantidad\_Disponible = Cantidad\_Disponible - cantidadVendida

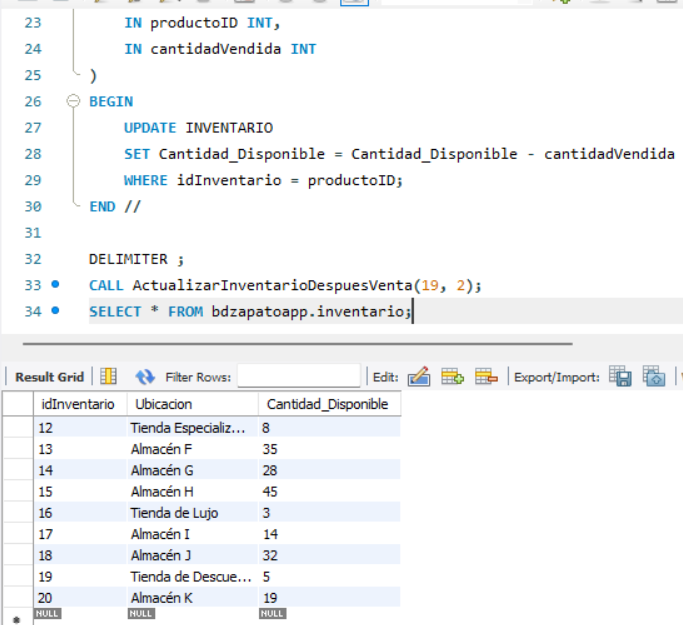
WHERE idInventario = productoID;

END //

DELIMITER ;

**Resultado:**

**Se le resto 2 al inventario 19**

****

* **Procedimiento para obtener el total de ventas por categoría**

DELIMITER //

CREATE PROCEDURE ObtenerTotalVentasPorCategoria()

BEGIN

SELECT C.Nombre AS Categoria, SUM(V.Total) AS TotalVentas

FROM VENTA V

JOIN PRODUCTO P ON V.Producto = P.Nombre

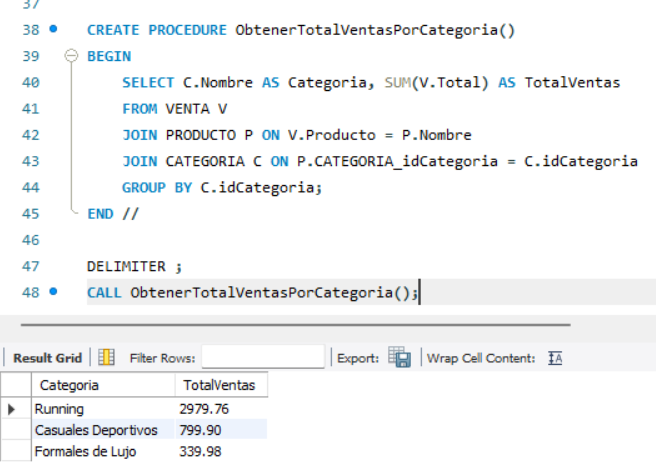
JOIN CATEGORIA C ON P.CATEGORIA\_idCategoria = C.idCategoria

GROUP BY C.idCategoria;

END //

DELIMITER ;

**Resultado:**

****

**FUNCIONES:**

* **Función para obtener el nombre completo de un empleado**

DELIMITER //

CREATE FUNCTION ObtenerNombreCompletoEmpleado(empleado\_id INT) RETURNS VARCHAR(100) DETERMINISTIC

BEGIN

DECLARE nombre\_completo VARCHAR(100);

SELECT CONCAT(Nombre1, ' ', Nombre2, ' ', Apellido1, ' ', Apellido2) INTO nombre\_completo

FROM EMPLEADO

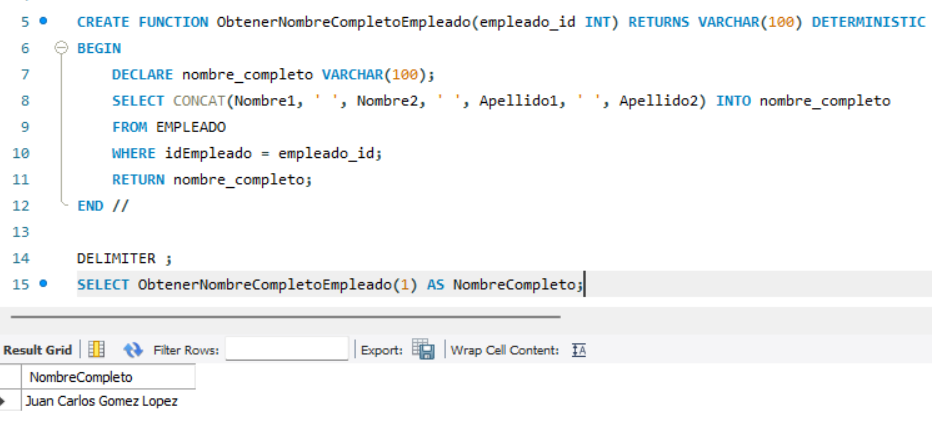
WHERE idEmpleado = empleado\_id;

RETURN nombre\_completo;

END //

DELIMITER ;

**Resultado:**



* **función para obtener el precio promedio de los productos en una categoría**

DELIMITER //

CREATE FUNCTION ObtenerPrecioPromedioCategoria(categoria\_id INT) RETURNS DECIMAL(10, 2) DETERMINISTIC

BEGIN

DECLARE precio\_promedio DECIMAL(10, 2);

SELECT AVG(Precio) INTO precio\_promedio

FROM PRODUCTO

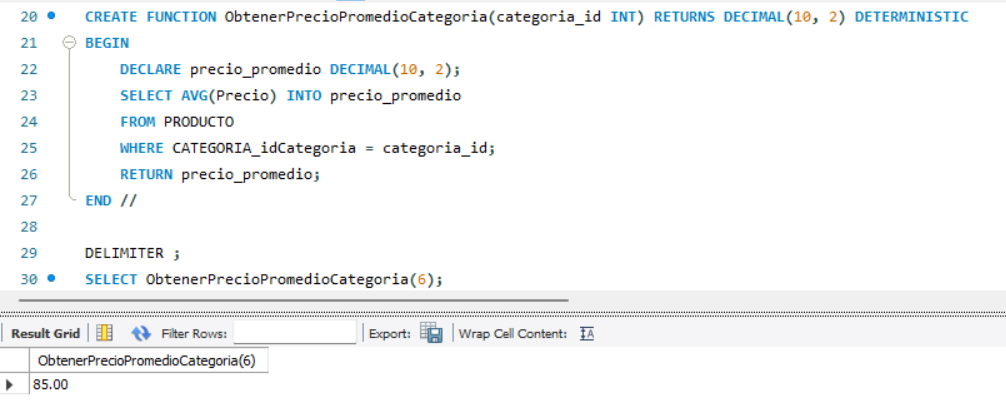
WHERE CATEGORIA\_idCategoria = categoria\_id;

RETURN precio\_promedio;

END //

DELIMITER ;

**Resultado:**



* **función para obtener el proveedor con más productos en inventario**

DELIMITER //

CREATE FUNCTION ObtenerProveedorMasProductos() RETURNS VARCHAR(50) DETERMINISTIC

BEGIN

DECLARE proveedor\_mas\_productos VARCHAR(50);

SELECT PROVEEDOR.Nombre INTO proveedor\_mas\_productos

FROM PRODUCTO

JOIN PROVEEDOR ON PRODUCTO.PROVEEDOR\_idProveedor = PROVEEDOR.idProveedor

GROUP BY PRODUCTO.PROVEEDOR\_idProveedor

ORDER BY COUNT(\*) DESC

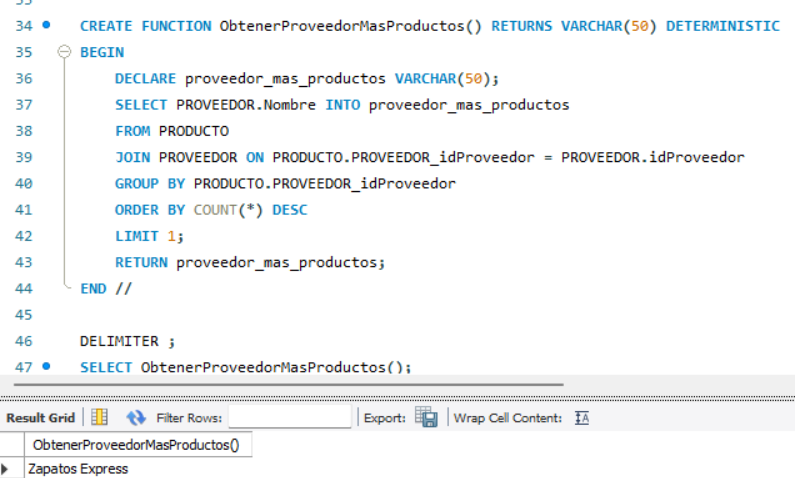
LIMIT 1;

RETURN proveedor\_mas\_productos;

END //

DELIMITER ;

**Resultado:**



**TRIGGERS:**

* **Disparador para registrar cambios en el precio de un producto**

DELIMITER //

CREATE TRIGGER RegistrarCambioPrecio

BEFORE UPDATE ON PRODUCTO

FOR EACH ROW

BEGIN

IF NEW.Precio != OLD.Precio THEN

INSERT INTO HISTORIAL\_CAMBIOS\_PRECIOS (idProducto, PrecioAnterior, PrecioNuevo)

VALUES (NEW.idProducto, OLD.Precio, NEW.Precio);

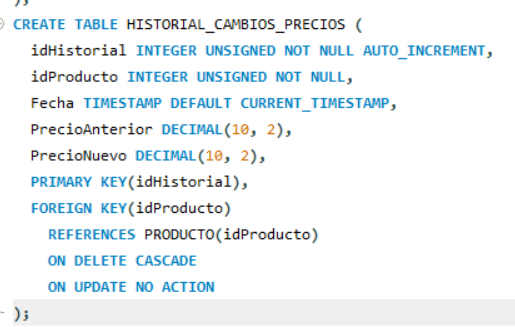
END IF;

END;

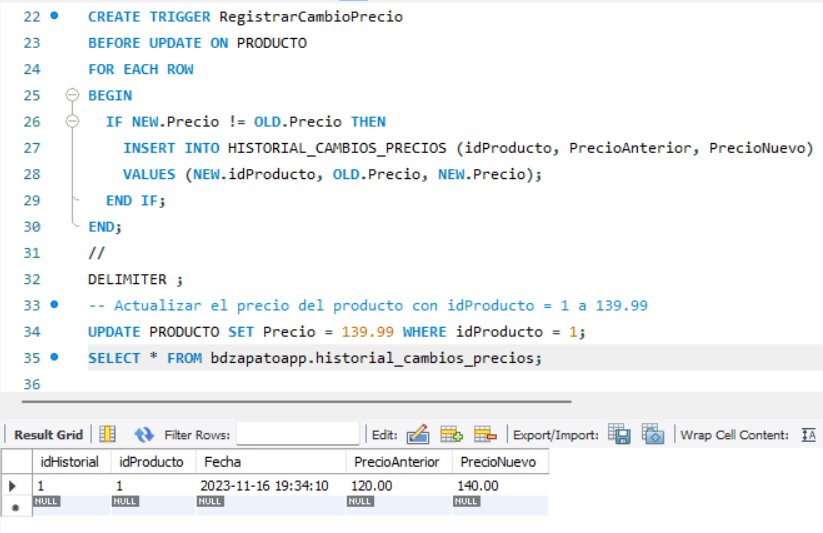
//

DELIMITER ;

Se crea la nueva tabla:



**Resultado:**



* **Disparador para registrar un historial de ventas**

DELIMITER //

CREATE TRIGGER RegistrarHistorialVentas

AFTER INSERT ON VENTA

FOR EACH ROW

BEGIN

INSERT INTO HISTORIAL\_VENTAS (idVenta, Fecha, Producto, Cantidad, Total)

VALUES (NEW.idVenta, NEW.Fecha, NEW.Producto, NEW.Cantidad, NEW.Total);

END;

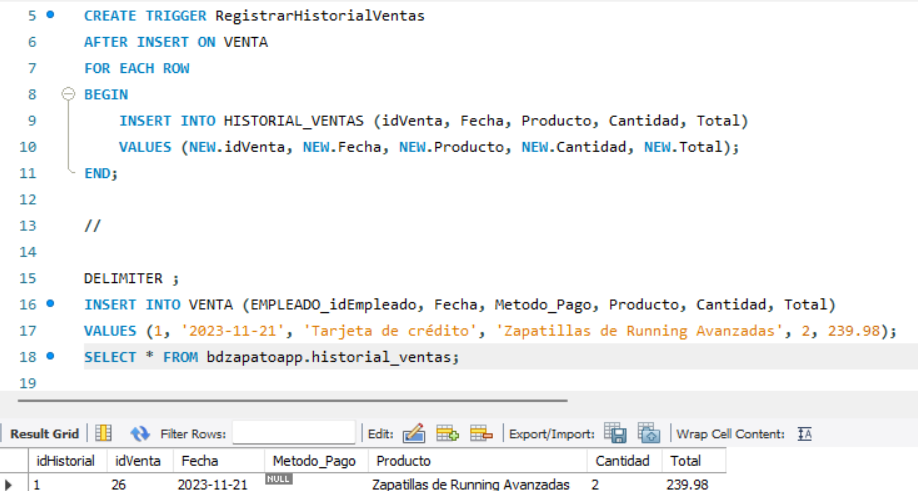
//

DELIMITER ;

**Se crea una nueva tabla:**



**Resultado:**



* **Disparador para actualizar automáticamente el total de ventas al insertar una nueva venta**

-- Crear el disparador para actualizar automáticamente el total de ventas

DELIMITER //

CREATE TRIGGER ActualizarTotalVentas

BEFORE INSERT ON VENTA

FOR EACH ROW

BEGIN

SET NEW.Total = NEW.Cantidad \* (SELECT Precio FROM PRODUCTO WHERE Nombre = NEW.Producto);

END;

//

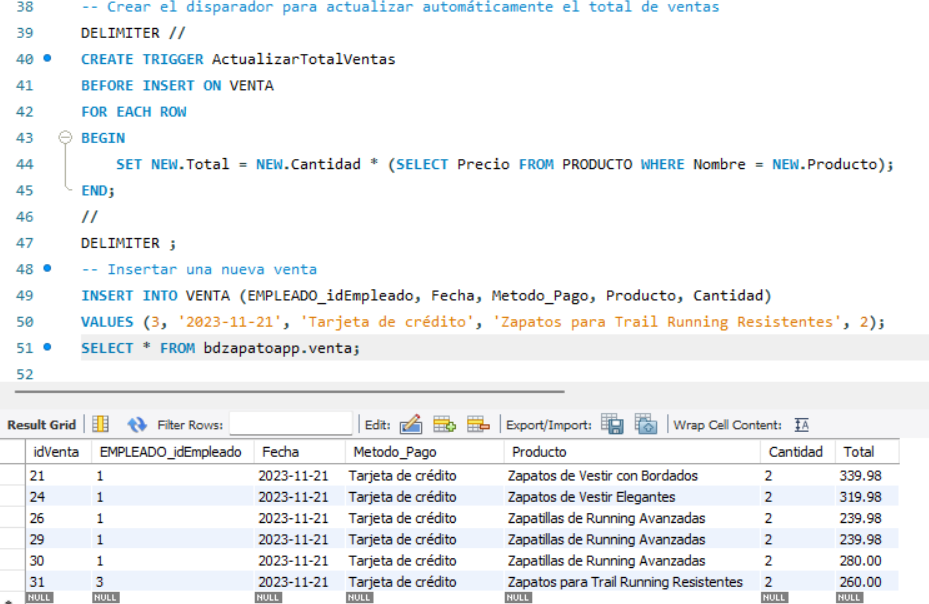
DELIMITER ;

**Resultado:**

**Antes:**

****

**Ahora:**

****