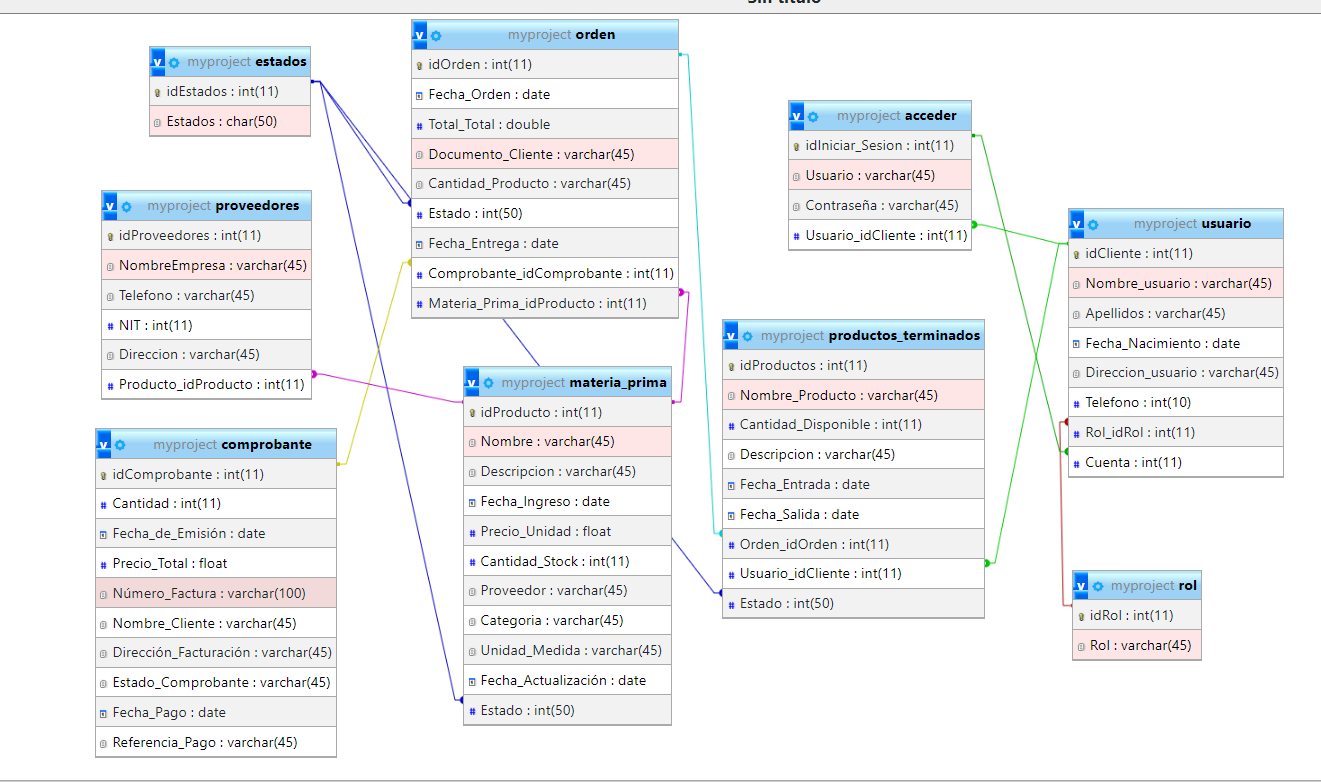
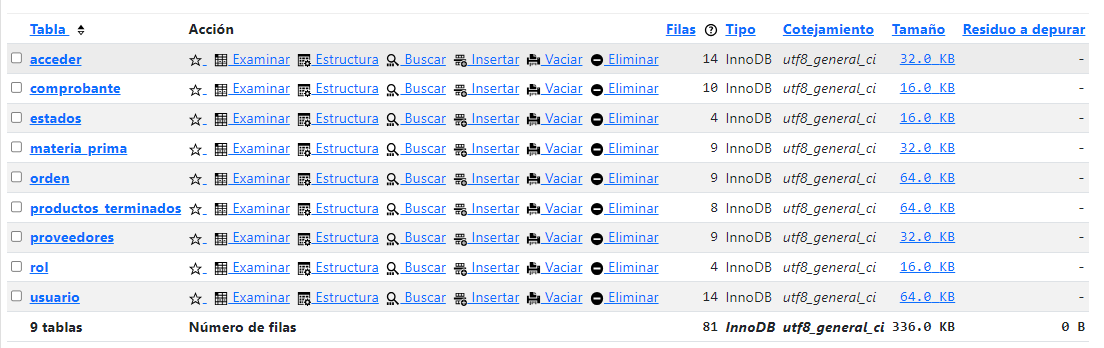
**BASE DE DATOS**

**CREATE DATABASE TexFashion;**

****

****

****

======================================================

**INSERT INTO**

**Tabla estados**

INSERT INTO estados (`Estados`)

VALUES

('Disponible'), ('No\_Disponible'), ('Habilitada'), ( 'No\_habilitada')

**Tabla usuario**

INSERT INTO usuario (`idCliente`,`Nombre\_usuario`,`Apellidos`,`Contraseña`,`Correo`,`Fecha\_Nacimiento`,`Direccion\_usuario`,`Telefono`,`Rol\_idRol`)

VALUES

(104,'Juliana Sutana', 'Gualteros', 'miClaveSecreta', 'JulianitaSutanita@ejemplo.com','19-07-2000', 'Carrera #89-90 sur', 231320194, 4),

(105, 'Aristobulo', 'Rodriguez', 'unaContraseñaNueva', 'AristobuloJasinto@ejemplo.com', '19-07-2000','Carrera #60-90 sur', 231451194,4),

(106, 'Edwardo', 'Camavingo', 'miClaveSecreta', 'EduardoCamavinga@ejemplo.com', '19-07-2000','Carrera #81-95 sur', 2611456194, 4),

( 107, 'Rodrigo', 'Depaul', 'miClaveSecreta', 'RodrigoDePaul@ejemplo.com', '19-07-2000','Carrera #20 sur', 235655194, 4),

(108, 'James', 'Rodriguez', 'miClaveSecreta','JamesRodriguez@ejemplo.com', '19-07-2000','Carrera #6590 sur', 25510194, 4),

(109, 'Richard', 'Rios', 'miClaveSecreta', 'RichardRioz@ejemplo.com', '19-07-2000','Carrera #20', 51056320194, 4),

(110, 'Sinadin', 'Zindan', 'miClaveSecreta', 'SinadinZidan@ejemplo.com', '19-07-2000','Carrera #54 sur', 23512094, 4),

(111, 'Lautora', 'Martinez', 'miClaveSecreta', 'LautaroMartinez@ejemplo.com', '19-07-2000','Carrera #290 sur', 55200194, 4),

(112, 'Bella', 'Porsh', 'miClaveSecreta', 'BellaPorsh@ejemplo.com', '19-07-2000','Carrera #89-92 sur', 25820094, 4),

(113, 'Aida', 'Merlano', 'miClaveSecreta', 'AidaVictoriaMerlano@ejemplo.com', '19-07-2000','Carrera #8-90 sur', 5522000194, 4);

===========================================================

**Tabla acceder**

INSERT INTO acceder (`Usuario`,`Contraseña`,`Usuario\_idCliente`)

VALUES

('maria.lopez@ejemplo.com','miClaveSecreta',100),

('david.gomez@ejemplo.com', 'miPass2023', 101),

('ana.martinez@ejemplo.com', 'claveSegura2024', 102),

('pedro.rodriguez@ejemplo.com', 'micontraseña123', 103);

=======================================================

**Tabla comprobante**

INSERT INTO comprobante

(`Cantidad`, `Fecha\_de\_Emisión`, `Precio\_Total`, `Número\_Factura`, `Nombre\_Cliente`, `Dirección\_Facturación`, `Estado\_Comprobante`, `Fecha\_Pago`, `Referencia\_Pago`)

VALUES

(10, '2024-07-22', 1200.00, 'FACT20240001', 'Juan Perez Lopez', 'Calle 123 Ciudad X', 'Pagado', '2024-07-22', '1234567890'),

(5, '2024-07-20', 500.00, 'FACT20240002', 'Maria Garcia Flores', 'Carrera 10 # 50-20', 'Pendiente', NULL, NULL),

(2, '2024-07-15', 250.00, 'FACT20240003', 'Pedro Sanchez Morales', 'Avenida Central s/n', 'Anulado', '2024-07-18', '1234567890'),

(15, '2024-07-10', 2100.00, 'FACT20240004', 'Ana Lopez Perez', 'Calle 5 # 30-15', 'Pagado', '2024-07-12', '9876543210'),

(3, '2024-07-05', 375.00, 'FACT20240005', 'David Gomez Rodriguez', 'Transversal 14 # 65-40', 'Pendiente', NULL, NULL);

===============================================================

**Tabla materia prima**

INSERT INTO materia\_prima (`idProducto`,`Nombre`,`Descripcion`,`Fecha\_Ingreso`,`Precio\_Unidad`,`Cantidad\_Stock`,`Proveedor`,`Categoria`,`Unidad\_Medida`,`Fecha\_Actualización`,`Estado\_Materia prima`)

VALUES

(101, 'Tela de algodón', 'Tela de algodón 100% natural para confección de camisas y blusas', '2024-07-22', 12.00, 50, 'Textiles del Sur', 'Telas', 'Metros', '2024-07-22', 'Activo'),

(102, 'Hilo de poliéster', 'Hilo de poliéster de alta resistencia para costura', '2024-07-20', 3.00, 25, 'Hilados y Cierres El Progreso', 'Materiales de costura', 'Conos', '2024-07-22', 'Activo'),

(103, 'Botones', 'Botones de plástico blanco para camisas', '2024-07-18', 0.50, 100, 'Accesorios de Ropa y Moda', 'Unidades', '2024-07-22', 'Activo'),

(104, 'Cremalleras', 'Cremalleras de nylon para pantalones y faldas', '2024-07-15', 2.50, 30, 'Cierres y Complementos Textiles', 'Unidades', '2024-07-22', 'Activo');

============================================================

**Tabla orden**

INSERT INTO orden ('Fecha\_Orden','Estado\_Orden','Total\_Total','Documento\_Cliente','Cantidad\_Producto','Fecha\_Entrega','Comprobante\_idComprobante','Materia\_Prima\_idProducto') VALUES ( '2024-07-23', 'Procesada', 50000.00, '987654321', 1, '2024-07-24', 6, 101 ), ( '2024-07-23', 'Procesada', 50000.00, '987654321', 1, '2024-07-24', 7, 102 ), ('2024-07-23','Pendiente',75000.00,'987654321',2,'2024-07-26',8,103), ('2024-07-22','Procesada',90000.00,'123456789',4,'2024-07-25',8, 104);  
==================================================================  
  
**Tabla proveedores**  
INSERT INTO proveedores (`idProveedores`,`Nombre Empresa`,`Telefono`,`NIT`,`Direccion`,`Producto\_idProducto`) VALUES

(001,'Telasnorma',3214911633,0001,'Calle 10 No. 9 - 78 Centro',101),

(002,'telasfabrica',3133310303,0002,'Carrera 56A No. 51 - 81',102),

(003,'Telasnew',3214915698,0003,'Carrera 22 No. 17-31',103),

(004,'Confecion',3214915645,0004,'Carrera 54 No. 68 - 80 Barrio el Prado',104);

==================================================================

**Tabla productos terminados**INSERT INTO productos\_terminados (`idProductos`,`Nombre\_Producto`,`Cantidad\_Disponible`,`Descripcion`,`Fecha\_Entrada`,`Fecha\_Salida`,`Estado`,`Orden\_idOrden`,`Usuario\_idCliente` ) VALUES (106, 'Camiseta deportiva', 15, 'Camiseta de algodón para hombre, color azul, talla L', '2024-07-23', NULL, 'Disponible',1 , 100); (107, 'Pantalón jeans', 12, 'Pantalón jeans para mujer, color azul oscuro, talla 38', '2024-07-22', NULL, 'Disponible', 2, 101); (108, 'Vestido floreado', 8, 'Vestido de verano para mujer, estampado floral, talla M', '2024-07-21', NULL, 'Disponible', 3, 102); (109, 'Zapatos deportivos', 10, 'Zapatos deportivos para hombre, color negro, talla 42', '2024-07-20', '2024-07-23', 'Vendido', 4, 103);

=================================================================

**Llaves foránea añadidas**

ALTER TABLE orden

ADD CONSTRAINT fk\_anios

FOREIGN KEY (Estado)

REFERENCES estados(idEstados);

ALTER TABLE materia\_prima

ADD CONSTRAINT fk\_anio

FOREIGN KEY (Estado)

REFERENCES estados(idEstados);

ALTER TABLE productos terminados

ADD CONSTRAINT fk\_ani

FOREIGN KEY (Estado)

REFERENCES estados(idEstados);

ALTER TABLE usuario

ADD CONSTRAINT fk\_a

FOREIGN KEY (Cuenta)

REFERENCES acceder(idIniciarSesion);

**CONSULTAS**

**Tabla usuarios**

Se realiza una consulta para saber los usuarios que tienen el rol de cliente

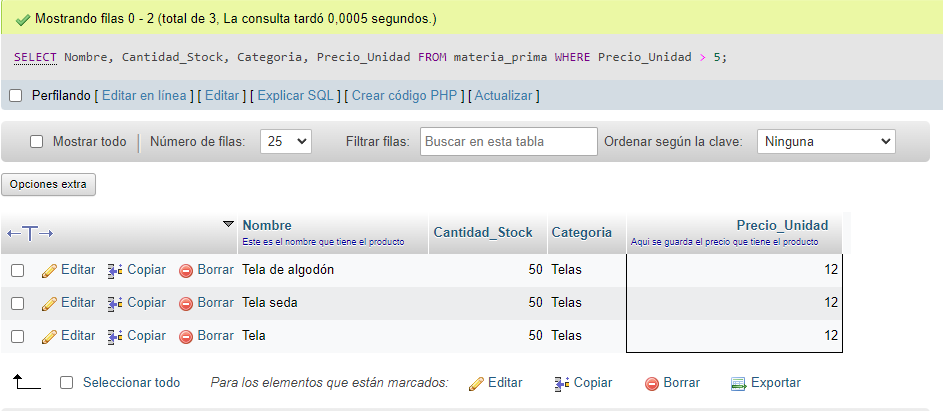
SELECT Nombre\_usuario, Apellidos, Rol\_idRol FROM usuario WHERE Rol\_idRol=4;



**Tabla materia prima**

Se realiza la consultas del precio de las unidades que sean mayores a 5

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) Nombre, Cantidad\_Stock, Categoria, Precio\_Unidad FROM materia\_prima WHERE Precio\_Unidad > 5;

**Tabla productos terminados****S**

Se realiza la consulta para ver el estado del producto

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) idProductos, Nombre\_Producto, Descripcion, Estado FROM productos\_terminados WHERE Estado = 'Disponible';



**Tabla productos terminados**

Se realiza la consulta para organizar de forma ascendente la cantidad disponible de los productos

SELECT Nombre\_Producto, Cantidad\_Disponible FROM productos\_terminados ORDER BY Cantidad\_Disponible ASC;



**Tabla orden**

Se utiliza la consulta para ver las fechas de entrega antes del 2024-07-25

SELECT \* FROM orden WHERE Fecha\_Entrega < '2024-07-25';

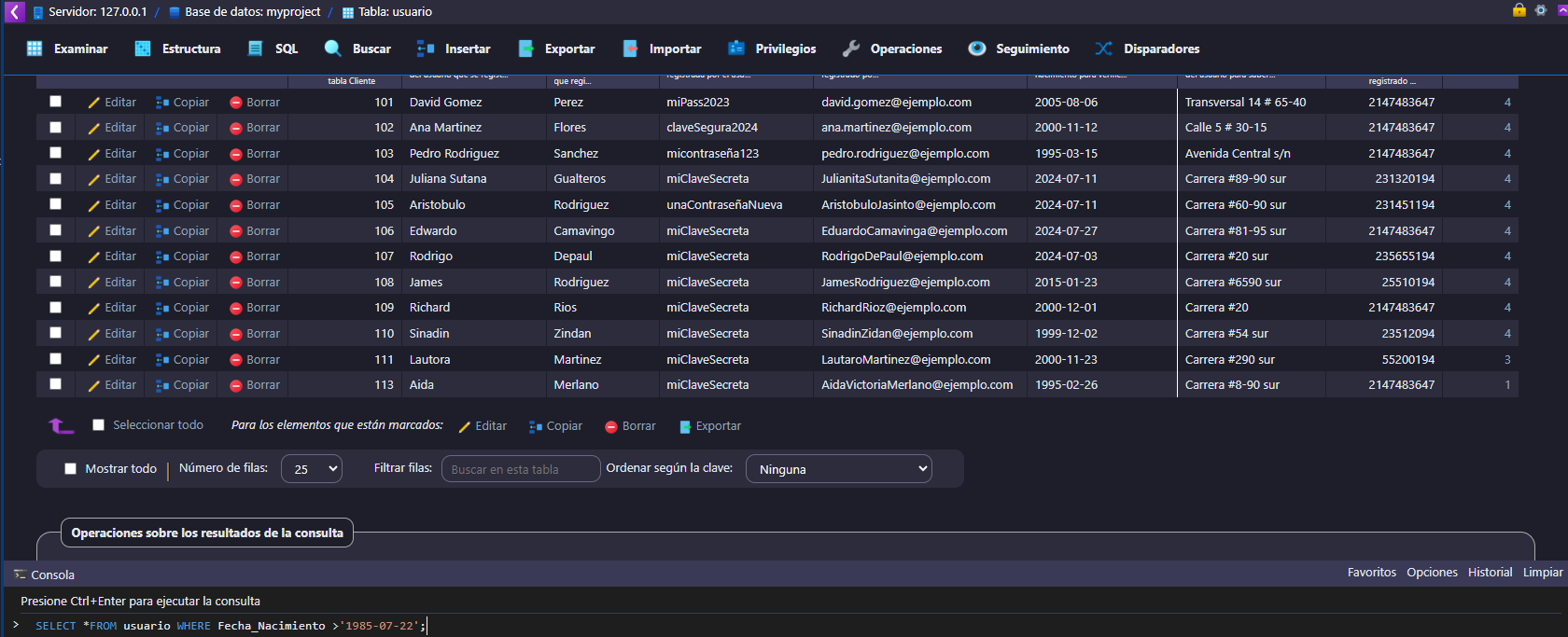


**Tabla usuario**

Descripción

Necesito que me llame a los usuarios registrados que tienen una fecha de nacimiento,  
mayor a 1985-07-22

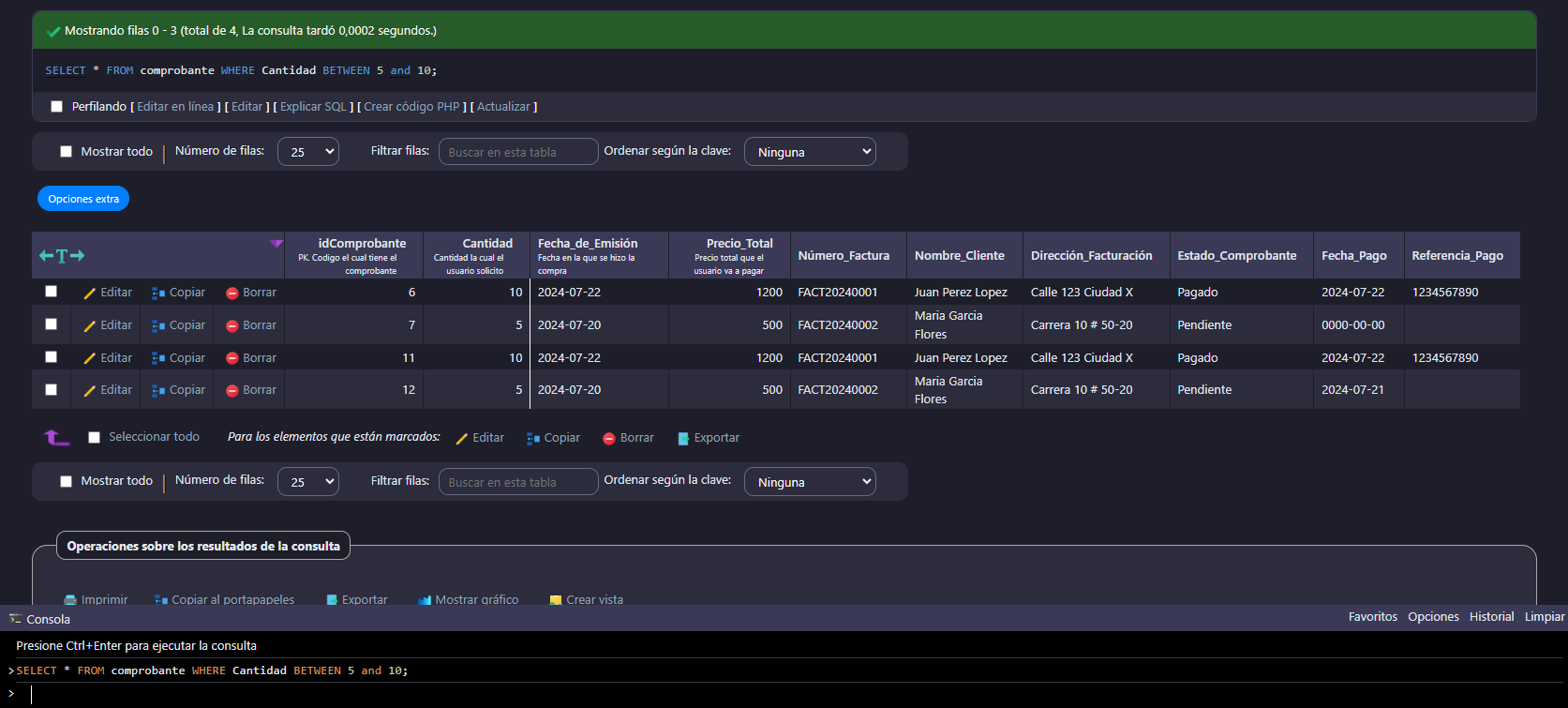
SELECT \*FROM usuario WHERE Fecha\_Nacimiento >'1985-07-22';



**Tabla comprobante**

Necesito que me llame la tabla comprobante y que me muestre la cantidad de 5 y 10.

**SELECT \* FROM comprobante WHERE Cantidad BETWEEN 5 and 10;**



**CONSULTAS MULTITABLA**

**Tabla orden**

Se realiza esta consulta para seleccionar la orden, fecha de realización el nombre de la materia prima solicitada en esta orden y el id del comprobante

SELECT o.idOrden, o.fecha\_orden, mp.nombre, c.comprobante\_idcomprobante FROM orden o JOIN comprobante c ON o.comprobante\_idComprobante = c.idComprobante JOIN materia\_prima mp ON mp.idProducto = o.Materia\_Prima\_idProducto;



**Tabla proveedores**

**S**e realiza la consulta para llamar el nombre de la empresa, el idproducto y el nombre del producto distribuido por el proveedor

SELECT P.NombreEmpresa, mp.idProducto, mp.nombre FROM proveedores P JOIN materia\_prima mp ON mp.idProducto= P.Producto\_idProducto;



**Tabla Productos\_terminados**

Se realiza la consulta para llamar el idproductos, el nombre del producto y la fecha de salida distribuido por productos terminados.

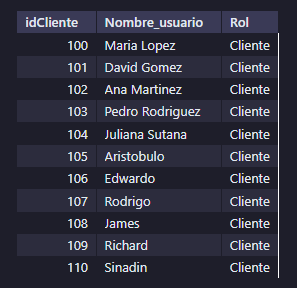
SELECT pt.idProductos, pt.Nombre\_Producto, pt.Fecha\_Salida FROM productos\_terminados pt JOIN proveedores P ON P.producto\_idProducto=pt.idProductos;

…………………………………………………………..  


**Tabla usuario**

Se realiza la consulta para llamar el id ciente y el nombre del usuario de la tabla usuario y de la tabla rol para saber el rol que tiene el usuario

SELECT u.idCliente, u.Nombre\_usuario, r.Rol FROM usuario u JOIN Rol r ON u.Rol\_idRol = r.idRol WHERE r.Rol = 'cliente';



**Tabla estados**

Se realiza la consulta para saber el estado de los productos

SELECT E.estados, mp.nombre, mp.estado, pt.Nombre\_Producto, pt.Estado FROM estados E JOIN materia\_prima mp ON E.idEstados = mp.Estado JOIN productos\_terminados pt ON E.idEstados= pt.Estado;



**Tabla Orden.  
S**e realiza la consulta para llamar el idOrden, la fecha\_Orden y el idProducto distribuido por la tabla orden.   
SELECT o.idOrden, o.Fecha\_Orden, mp.idProducto FROM orden o JOIN materia\_prima mp ON o.Materia\_Prima\_idProducto= mp.idProducto;



**Triggers**

**Trigger BEFORE**

El trigger se activa antes de que se realice una operación de inserción en la tabla comprobante. Su función principal es validar el valor del campo Precio\_Total antes de que el nuevo registro sea insertado en la base de datos.

DELIMITER $$

CREATE TRIGGER validar\_precio\_total\_before\_insert

BEFORE INSERT ON comprobante

FOR EACH ROW

BEGIN

IF NEW.Precio\_Total < 0 THEN

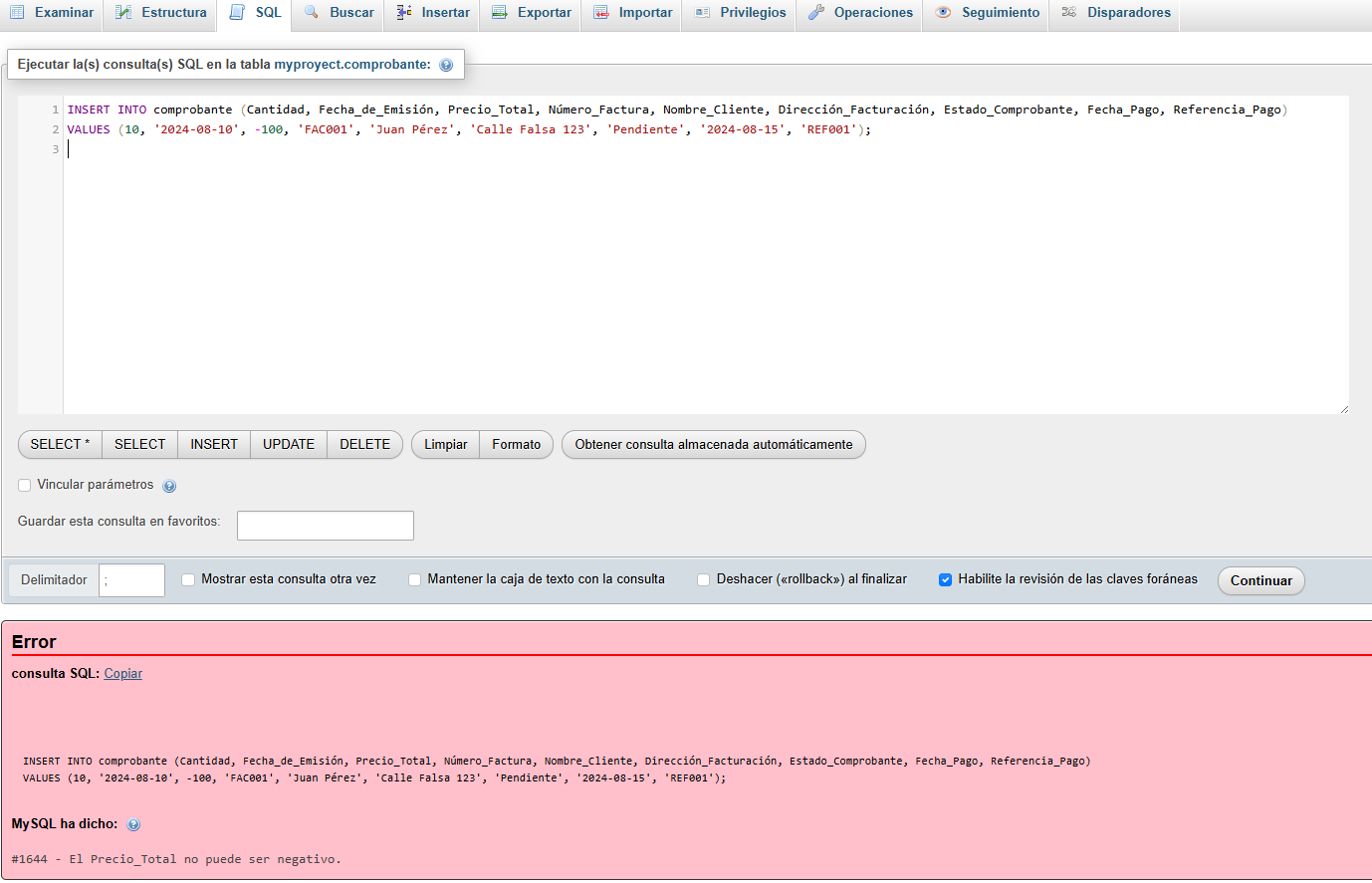
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'El Precio\_Total no puede ser negativo.';

END IF;

END$$

DELIMITER ;

**Prueba TRIGGER**



**Trigger AFTER**

El trigger AFTER INSERT que hemos definido se utiliza para registrar automáticamente en una tabla de log los detalles de cada nuevo registro que se inserta en la tabla roles. Esto se hace para mantener un historial o registro de las operaciones realizadas, lo cual puede ser útil para auditorías, análisis de cambios o seguimiento.

DELIMITER $$

CREATE TRIGGER log\_roles\_after\_insert

AFTER INSERT ON rol

FOR EACH ROW

BEGIN

INSERT INTO log\_roles (idRol, Rol)

VALUES (NEW.idRol, NEW.Rol);

END$$

DELIMITER ;

**Creación tabla Log roles**

CREATE TABLE log\_roles (

id INT AUTO\_INCREMENT PRIMARY KEY,

idRol INT(11),

Rol VARCHAR(45),

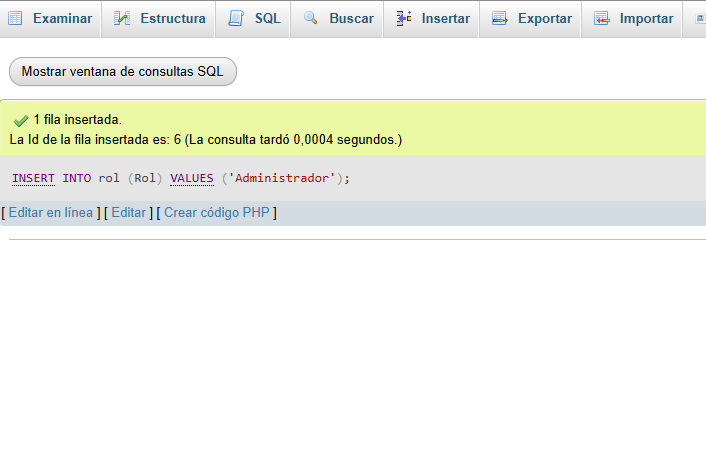
Fecha\_Log TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

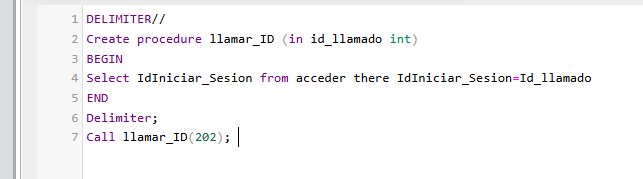
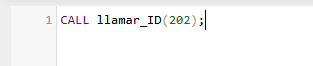
**PRUEBA TRIGGER**

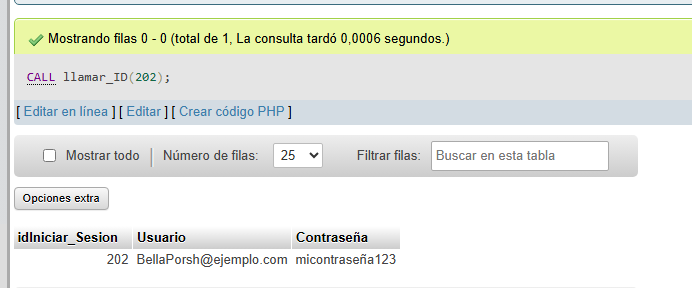
INSERT INTO rol (Rol)

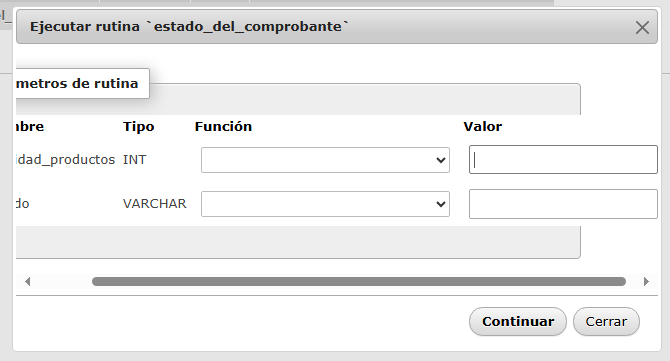
VALUES ('Administrador');



**Procedimientos con parámetros.**

Aqui lo que hice fue crear el procedimiento e cual nombre “llamar\_ID” Este procedimiento busca un registro específico en una tabla utilizando un ID proporcionado como entrada. Si encuentra una coincidencia, retorna un valor asociado a ese registro. **  
  
**

****

****

**Segundo procedimiento con parámetros**

**Funcion**

Pongo un delimitador y dentro del limitador hago mi función le pongo un nombre a la función declaro lo que va a recibir la función y declaro lo que va a retornar la función.

Que en este caso de claro que va a recibir un varchar y va a retornar un entero, Empiezo la función declarando en donde voy a guardar las veces que se repite la contraseña. Luego le digo que me cuente y que guarde lo que cuente dentro de la variable declarada.

Luego selecciono mi tabla y le digo que busque en la columna contraseña la contraseña que sea igual a la que ingresó el usuario.

DELIMITER //

create function contraseña\_repetida(contraseña\_usuario varchar(50)) RETURNS INT

BEGIN

DECLARE veces\_repetidas ins;

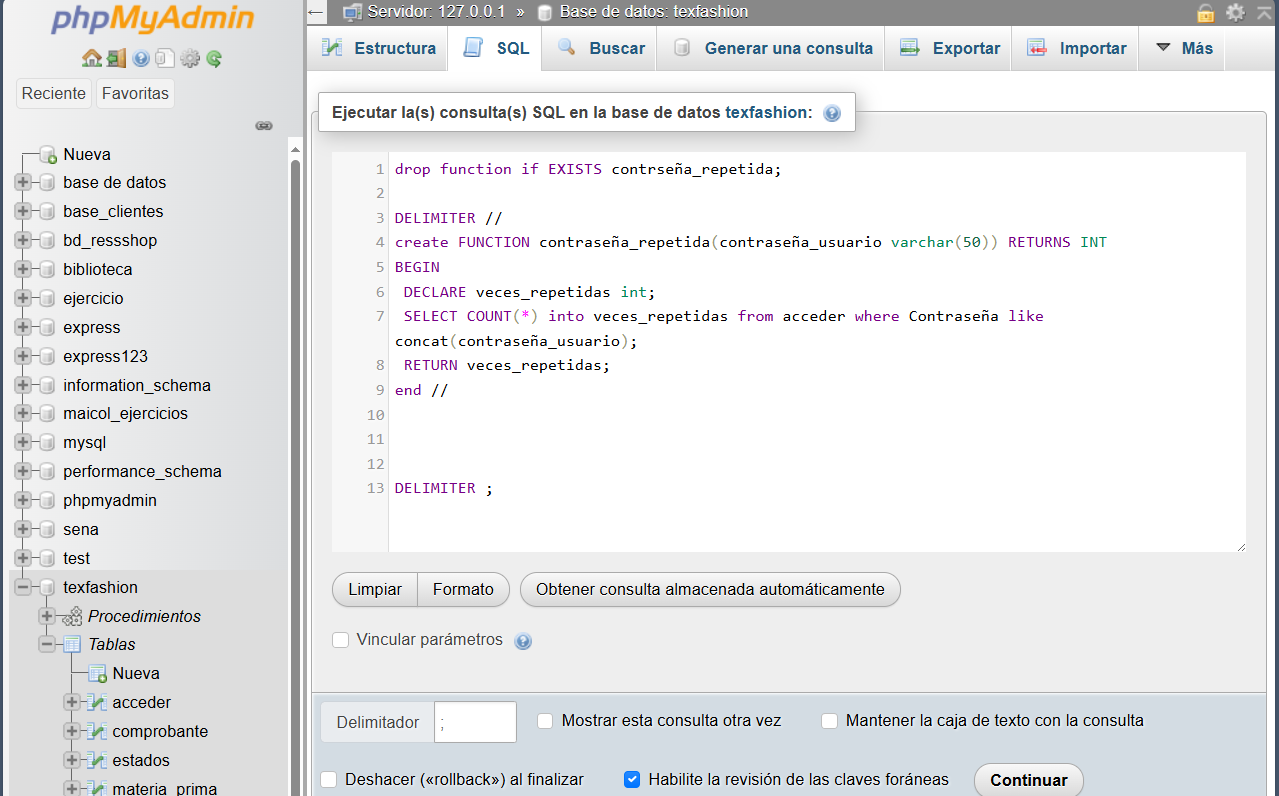
SELEC COUNT (\*) into veces\_repetidas from acceder where Contraseña like

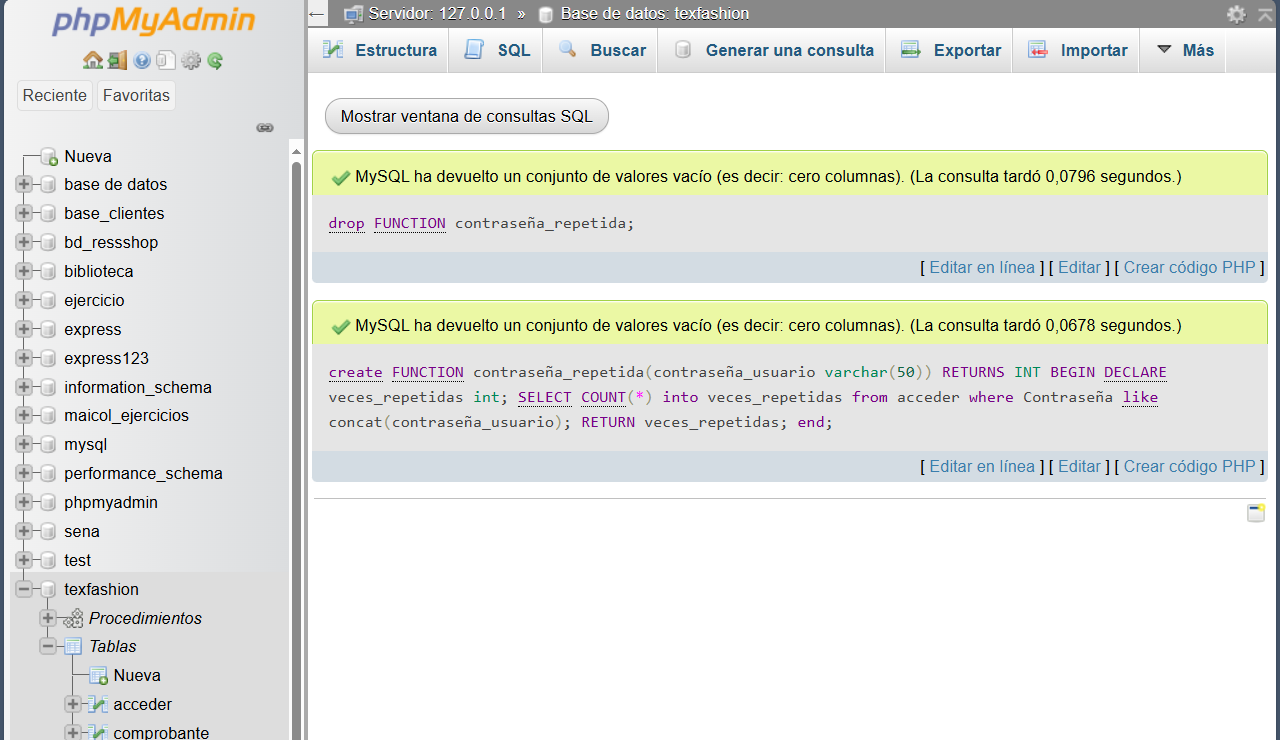
concat (contraseña\_usuario);

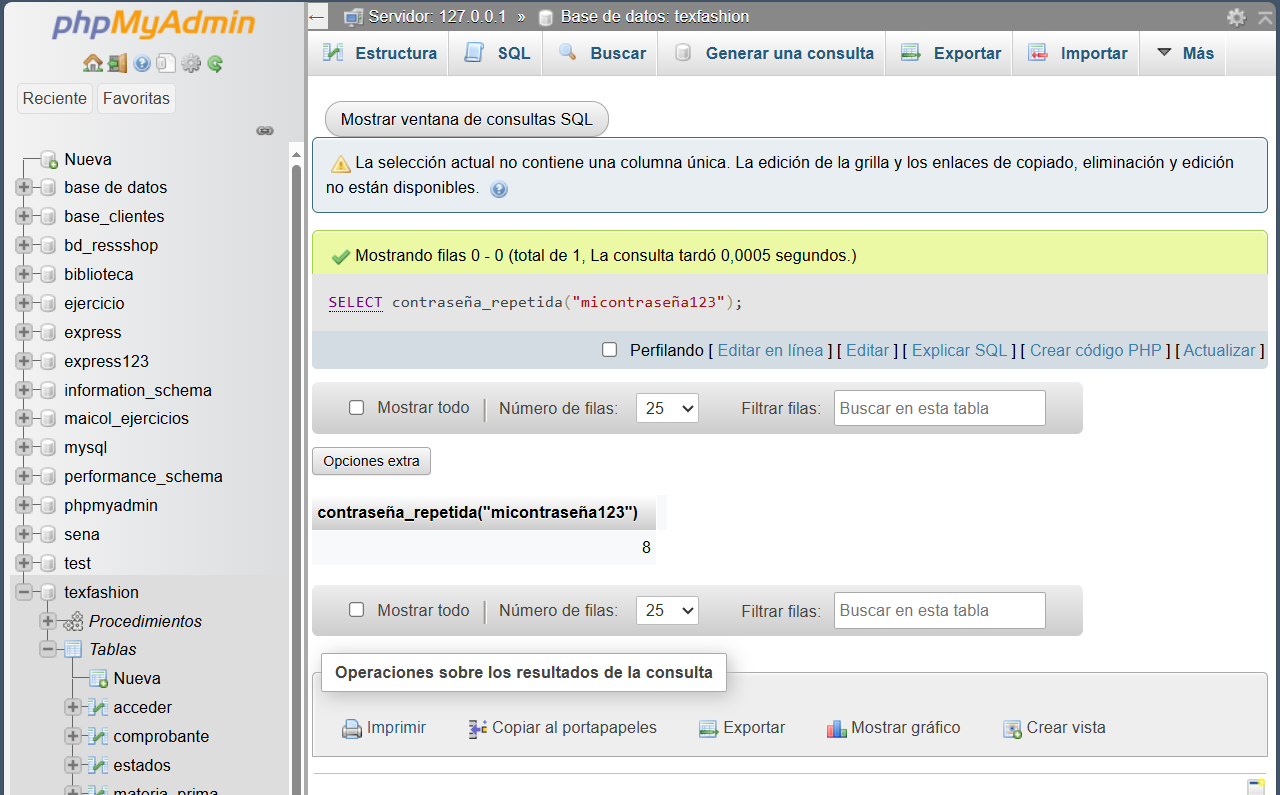
RETURN veces\_repetuidas;

end //

DELIMITER ;







**Encriptación de contraseña**

Se actualizaron los datos ya insertados

**UPDATE acceder**

**SET Contraseña = SHA2(Contraseña, 256);**

**Se verificó insertando nuevos datos**

**DELIMITER //**

**CREATE TRIGGER before\_insert\_Iniciar\_Sesion**

**BEFORE INSERT ON acceder**

**FOR EACH ROW**

**BEGIN**

**SET NEW.Contraseña = SHA2(NEW.Contraseña, 256);**

**END;**

**//**

**DELIMITER ;**

**DELIMITER //**

**CREATE TRIGGER before\_update\_Iniciar\_Sesion**

**BEFORE UPDATE ON Iniciar\_Sesion**

**FOR EACH ROW**

**BEGIN**

**SET NEW.Contraseña = SHA2(NEW.Contraseña, 256);**

**END;**

**//**

**DELIMITER ;**

**DELIMITER $$**

**CREATE TRIGGER EncryptPasswordBeforeInsert**

**BEFORE INSERT ON Acceder**

**FOR EACH ROW**

**BEGIN**

**SET NEW.Contraseña = SHA2(NEW.Contraseña, 256);**

**END$$**

**CREATE TRIGGER EncryptPasswordBeforeUpdate**

**BEFORE UPDATE ON Acceder FOR EACH ROW**

**BEGIN**

**SET NEW.Contraseña = SHA2(NEW.Contraseña, 256);**

**END$$**

**DELIMITER ;**

**Desencriptar**

**-- Desencriptar una contraseña**

**SET @key = '7871e134672760675191729eb098253975e09d30d8f9a**

**'; -- Debe ser la misma clave usada para encriptar**

**SELECT Usuario, AES\_DECRYPT(Contraseña, @key) AS Contraseña**

**FROM acceder**

**WHERE Usuario = 'AidaVictoriaMerlano@ejemplo.com';**