Creación de tablas:

1. Tabla Rol:

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]	
idRol tipoRol	
Operaciones sobre los resultados de la consulta	
Crear vista	

 $\label{eq:creater} \textbf{CREATE TABLE Rol (idRol int (10) NOT NULL AUTO_INCREMENT , tipoRol varchar (25) , PRIMARY KEY (idRol));}$

2. Tabla Usuario:

(Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]		
	Tipo_Documento Num_Documento Nombres Apellidos Teléfono Correo RolidRol LoginCorreo EstadoCodigoEstado GeneroidGenero		
-[Operaciones sobre los resultados de la consulta		
	™ Crear vista		

CREATE TABLE Usuario (Tipo_Documento varchar(30), Num_Documento int(15) NOT NULL AUTO_INCREMENT, Nombres varchar(25), Apellidos varchar(30), Teléfono int(10), Correo varchar(45), RolidRol int(10) NOT NULL, LoginCorreo varchar(25) NOT NULL, EstadoCodigoEstado int(10) NOT NULL, GeneroidGenero int(10) NOT NULL, PRIMARY KEY (Num_Documento));

3. Tabla categoría:

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]		
CodigoCategoría Nombre GeneroidGenero		
Operaciones sobre los resultados de la consulta		
💽 Crear vista		

<u>CREATE</u> TABLE Categoria (CodigoCategoría int(10) NOT NULL AUTO_INCREMENT, Nombre varchar(45), GeneroidGenero int(10) NOT NULL, PRIMARY KEY (CodigoCategoría));

4. Tabla producto:

	Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
ı	CodigoProducto Nombre Precio Stock IVA Imagen CategoriaCodigoCategoria EstadoCodigoEstado Descripcion MarcasCodigoMarca TallasCodigoTallas
	Operaciones sobre los resultados de la consulta
	R Crear vista

CREATE TABLE Producto (CodigoProducto int(12) NOT NULL AUTO_INCREMENT, Nombre varchar(25) , Precio double , Stock int(10) , IVA int(10) , Imagen varchar(25) , CategoriaCodigoCategoría int(10) NOT NULL , EstadoCodigoEstado int(10) NOT NULL , Descripcion varchar(45) , MarcasCodigoMarca int(10) NOT NULL , TallasCodigoTallas int(10) NOT NULL , PRIMARY KEY (CodigoProducto));

5. Tabla de ticket de ingreso:

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]		
	idTicketIngreso FechaIngresoProducto PrecioUnitario PrecioTotal	
	Operaciones sobre los resultados de la consulta	
	Crear vista	

CREATE TABLE `Ticket Ingreso` (idTicketIngreso int(12) NOT NULL AUTO_INCREMENT, FechaIngresoProducto date, PrecioUnitario double, PrecioTotal double, PRIMARY KEY (idTicketIngreso));

6. Tabla detalle de ingreso:



CREATE TABLE DetalleIngreso (ProductoCodigoProducto int(12) NOT NULL, `Ticket SalidaidTicketIngreso` int(12) NOT NULL, NombreProducto varchar(25), Cantidad int(10), Descripción varchar(45), PrecioIngreso double, PRIMARY KEY (ProductoCodigoProducto, `Ticket SalidaidTicketIngreso`));

7. Tabla de ticket de salida:

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actu	alizar]
IdTicketSalida FechaSalida PrecioTotal	
Operaciones sobre los resultados de la consulta	
CREATE TABLE `Ticket Salida` (IdTicketSalida int(12) NOT NULL AUTO_INCREMENT date , PrecioTotal double , PRIMARY KEY (IdTicketSalida));	, FechaSalida
8. Tabla detalle de salida:	
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]	
ProductoCodigoProducto Ticket SalidaldTicketSalida NombreProducto Descricpción Cantidad	d ValorUnitario
Operaciones sobre los resultados de la consulta	
CREATE TABLE `Detalle Salida` (ProductoCodigoProducto int(12) NOT NULL, `Tick SalidaldTicketSalida` int(12) NOT NULL, NombreProducto varchar(25), Descricpcid Cantidad int(10), ValorUnitario double, PRIMARY KEY (ProductoCodigoProducto, `Tick SalidaldTicketSalida`));	ón varchar(45) ,
9. Tabla login:	
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actu	alizar]
Correo Password	
Operaciones sobre los resultados de la consulta	
CREATE TABLE Login (Correct varieties) NOT NULL Password varhinary (255) NOT	NIIII IINIOUE

CREATE TABLE Login (Correo varchar(25) NOT NULL, Password varbinary(255) NOT NULL UNIQUE, PRIMARY KEY (Correo));

10. Tabla estado:



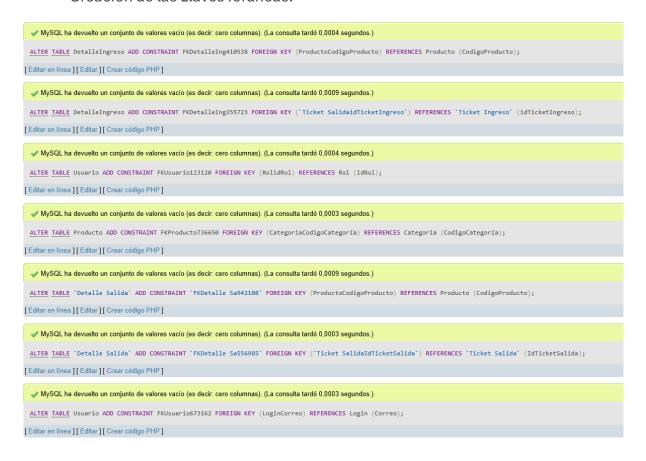
CREATE TABLE Tallas (CodigoTallas int(10) NOT NULL AUTO_INCREMENT, Tallas int(8), PRIMARY KEY (CodigoTallas));

13. Tabla Genero:



CREATE TABLE Genero (idGenero int(10) NOT NULL AUTO_INCREMENT, Nombre varchar(25), PRIMARY KEY (idGenero));

Creación de las Llaves foraneas:



```
ALTER TABLE Usuario ADD CONSTRAINT FKUsuario280302 FOREIGN KEY (EstadoCodigoEstado) REFERENCES Estado (CodigoEstado);
[Editar en línea][Editar][Crear código PHP]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0003 segundos.)

 ALTER TABLE Producto ADD CONSTRAINT FKProducto323712 FOREIGN KEY (EstadoCodigoEstado) REFERENCES Estado (CodigoEstado);
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
 ALTER TABLE Producto ADD CONSTRAINT FKProducto469343 FOREIGN KEY (MarcasCodigoMarca) REFERENCES Marcas (CodigoMarca);
[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0003 segundos.)

 ALTER TABLE Producto ADD CONSTRAINT FKProducto698452 FOREIGN KEY (TallasCodigoTallas) REFERENCES Tallas (CodigoTallas);
[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0008 segundos.)

 ALTER TABLE Categoria ADD CONSTRAINT FKCategoria900753 FOREIGN KEY (GeneroidGenero) REFERENCES Genero (idGenero);
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
 ALTER TABLE Usuario ADD CONSTRAINT FKUsuario65594 FOREIGN KEY (GeneroidGenero) REFERENCES Genero (idGenero);
[Editar en línea] [Editar] [Crear código PHP]
```

ALTER TABLE DetalleIngreso ADD CONSTRAINT FKDetalleIng410538 FOREIGN KEY (ProductoCodigoProducto) REFERENCES Producto (CodigoProducto);

ALTER TABLE DetalleIngreso ADD CONSTRAINT FKDetalleIng255723 FOREIGN KEY (`Ticket SalidaidTicketIngreso`) REFERENCES `Ticket Ingreso` (idTicketIngreso);

ALTER TABLE Usuario ADD CONSTRAINT FKUsuario123120 FOREIGN KEY (RolidRol) REFERENCES Rol (idRol);}

 $ALTER\,TABLE\,Producto\,ADD\,CONSTRAINT\,FK Producto736650\,FOREIGN\,KEY\,(CategoriaCodigoCategoria)\\ REFERENCES\,Categoria\,(CodigoCategoria);$

ALTER TABLE `Detalle Salida` ADD CONSTRAINT `FKDetalle Sa942108` FOREIGN KEY (ProductoCodigoProducto) REFERENCES Producto (CodigoProducto);

ALTER TABLE `Detalle Salida` ADD CONSTRAINT `FKDetalle Sa556985` FOREIGN KEY (`Ticket SalidaldTicketSalida`) REFERENCES `Ticket Salida` (IdTicketSalida);

 ${\tt ALTER\,TABLE\,Usuario\,ADD\,CONSTRAINT\,FKUsuario\,673162\,FOREIGN\,KEY\,(LoginCorreo)\,REFERENCES\,Login\,(Correo);}$

ALTER TABLE Usuario ADD CONSTRAINT FKUsuario 280302 FOREIGN KEY (Estado Codigo Estado) REFERENCES Estado (Codigo Estado);

ALTER TABLE Producto ADD CONSTRAINT FKProducto323712 FOREIGN KEY (EstadoCodigoEstado) REFERENCES Estado (CodigoEstado);

ALTER TABLE Producto ADD CONSTRAINT FKProducto469343 FOREIGN KEY (MarcasCodigoMarca) REFERENCES Marcas (CodigoMarca);

 $\label{thm:constraint} \mbox{ALTER TABLE Producto ADD CONSTRAINT FKProducto 698452 FOREIGN KEY (Tallas Codigo Tallas) REFERENCES Tallas (Codigo Tallas);} \\$

ALTER TABLE Categoria ADD CONSTRAINT FKCategoria 900753 FOREIGN KEY (Generoid Genero) REFERENCES Genero (id Genero);

ALTER TABLE Usuario ADD CONSTRAINT FKUsuario65594 FOREIGN KEY (GeneroidGenero) REFERENCES Genero (idGenero);

Inserción de parametros

Genero:

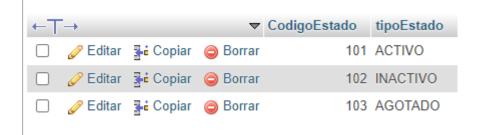
1	Masculino
2	Femenino
3	Genero no Binario
4	Genero fluido
5	Unisex



<u>insert</u> into genero (idGenero, Nombre) values (1, 'Masculino'), (2, 'Femenino'), (3, 'Genero no binario'), (004, 'Genero fluido'), (5, 'Unisex');

Estado:

101	ACTIVO
102	INACTIVO



 $\underline{insert}\ into\ estado\ (CodigoEstado,\ tipoEstado)\ VALUES\ (101, 'ACTIVO'), (102, 'INACTIVO'),\ ('103',\ 'AGOTADO');$

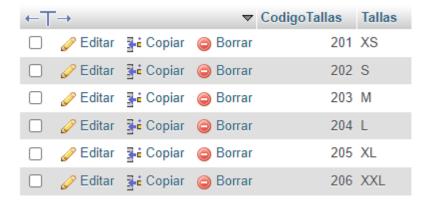
Categoría:

121	Conjuntos
122	Busos
123	Sudaderas
124	Camisas
125	Chaquetas
126	pantalonetas



Tallas:

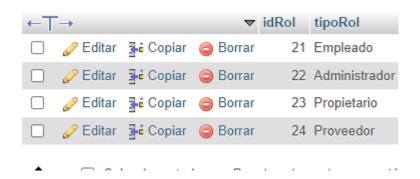
201	XS
202	S
203	М
204	L
205	XL
206	XXL



INSERT INTO Tallas (CodigoTallas, Tallas) values (201, 'XS'), (202, 'S'), (203, 'M'), (204, 'L'), (205, 'XL'), (206, 'XXL');

Rol:

21	Empleado
22	Administrador
23	Propietario
24	Proveedor



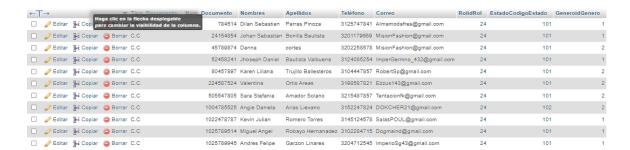
Marcas:

30	DK
31	Alma
32	Dogma IND
33	ImperioSg
34	Impermeables Gernimo
35	Paul Salas
36	Robert sp
37	Ezzus
38	Tentacion Sudaderas
39	Mision Fashion
40	Adinkra Ps

\leftarrow T			∇	CodigoMarca	Nombre
	<i></i> €ditar	≩ ≟ Copiar	Borrar	30	DK
	Ø Editar	≩ Copiar	Borrar	31	Alma
	<i></i> €ditar	≩	Borrar	32	Dogma IND
	Ø Editar	≩ Copiar	Borrar	33	ImperioSg
	<i></i> Editar	≩	Borrar	34	Impermeables Gernimo
	Editar	≩ ≟ Copiar	Borrar	35	Paul Salas
	<i></i> € Editar	≩ Copiar	Borrar	36	Robert sp
	Editar	≩ ≟ Copiar	Borrar	37	Ezzus
	<i></i> € Editar	≩ Copiar	Borrar	38	Tentacion Sudaderas
	Ø Editar	≩ ≟ Copiar	Borrar	39	Mision Fashion
	Editar	≩ Copiar	Borrar	40	Adinkra Ps

insert into marcas (CodigoMarca, Nombre) VALUES (30,'DK'),(31,'Alma'),(32,'Dogma IND'),(33,'ImperioSg'),(34,'Impermeables Gernimo '), (35,'Paul Salas'),(36,'Robert sp'),(37,'Ezzus'),(38,'Tentacion Sudaderas'),(39,'Mision Fashion'),(40,'Adinkra Ps');

Usuarios:



insert into usuario

(Tipo_Documento,Num_Documento,Nombres,Apellidos,Teléfono,Correo,RolidRol,LoginNum_Documento,EstadoCodigoEstado,GeneroidGenero) VALUES

('C.C',1025789514, 'Miguel Angel', 'Robayo Hernanadez', 3102284715, 'Dogmaind@gmail.com', 24,1025789514, 101, 1),

('C.C',1004785525, 'Angie Daniela', 'Arias Lievano', 3152247824, 'DOKCHER21@gmail.com', 24,1004785525, 102, 2),

('PPT',784514, 'Dilan Sebastian', 'Parras Pinoza',3125747841, 'Almamodafres@gmail.com', 24,784514, 101, 1),

('C.C',1025789945, 'Andres Felipe', 'Garzon Linares',3204712545, 'ImperioSg43@gmail.com', 24,1025789945, 101, 1),

('C.C',52458241, 'Jhoseph Daniel', 'Bautista Valbuena',3124085254, 'ImperGermno_432@gmail.com', 24,52458241, 101, 1),

('C.C',1022478787, 'Kevin Julian', 'Romero Torres',3145124578, 'SalasPOUL@gmail.com', 24,1022478787, 101, 1),

('C.C',80457897, 'Karen Liliana', 'Trujillo Ballesteros',3104447857, 'RobertSp@gmail.com', 24,80457897, 101, 2),

('C.C',224587524, 'Valentina', 'Ortis Areas',3198587821, 'Ezzus143@gmail.com', 24,224587524, 101, 2).

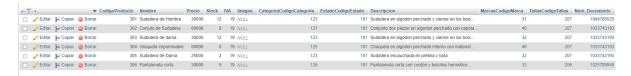
('C.C',505547805, 'Sara Stefania', 'Amador Solano',3215487857, 'Tentacionfk@gmail.com', 24,505547805, 101, 2),

('C.C',24154854, 'Johan Sebastian', 'Bonilla Bautista ',3201179669, 'MisionFashion@gmail.com', 24,24154854, 101, 1),

('C.C',45789874, 'Danna', 'cortes',3202258578, 'MisionFashion@gmail.com', 24,45789874, 101, 2)

('C.C',1033743193, 'Kevin Santiago ', 'Pulido Escobar ', 3192044487, 'KevinPulido0802@gmail.com', 21, 101, 1), ('C.C',80461462, 'Jonny Duverney', 'Pulido Monroy ', 3197875874, 'Jonny_duverney1973@hotmail.com.com', 21, 101, 1), ('C.C',52747225, 'Viviana ', 'Escobar Cortes', 3168524776, 'CortesViviana086@gmail.com', 22, 101, 2), ('C.C',51964869, 'Laureana ', 'Cortes Laguna ', 3104637195, 'Lauracortes343@gmail.com', 23, 101, 2)

Productos:



INSERT into producto

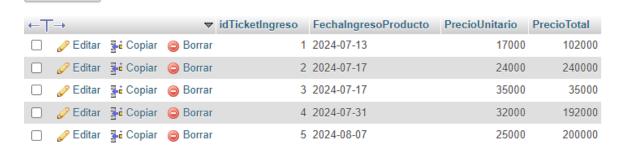
(CodigoProducto,Nombre,Precio,Stock,IVA,CategoriaCodigoCategoría,EstadoCodigoEstado,Descrip cion,MarcasCodigoMarca,TallasCodigoTallas,Num_Documento) VALUES (301, 'Sudadera de Hombre','30000',12,19,123,101,'Sudadera en algodon perchado y cierres en los bolsillos', 32,207,1033743193), (302, 'Conjuto de Sudadera ','60.000',8,19,121,101,'Conjunto dos piezas en algodon perchado con capota remobible', 40,207,1033743193);

Login;



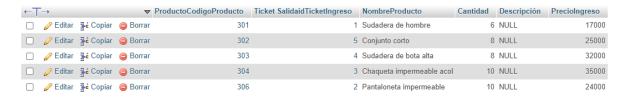
insert into login (Num_Documento,Password) VALUES (51964869,'Laureana123'),(52747225,'Viviana1972');

Ticket de Ingreso:



INSERT INTO `ticket_ingreso` (`idTicketIngreso`, `FechaIngresoProducto`, `PrecioUnitario`, `PrecioTotal`) VALUES ('1', '2024-07-13', '17000', '102000'), ('2', '2024-07-17', '24000', '240000'), ('3', '2024-07-17', '35000', '35000'), ('4', '2024-07-31', '32000', '192000'), ('5', '2024-08-07', '25000', '200000'):

Detalle Ingreso:



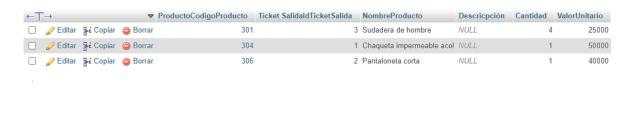
INSERT INTO `detalle_ingreso` (`ProductoCodigoProducto`, `Ticket SalidaidTicketIngreso`, `NombreProducto`, `Cantidad`, `Descripción`, `PrecioIngreso`) VALUES ('301', '1', 'Sudadera de hombre', '6', 'NULL', '17000'), ('306', '2', 'Pantaloneta impermeable', '10', 'NULL', '24000'), ('304', '3', 'Chaqueta impermeable acolchonada', '10', 'NULL', '35000'), ('303', '4', 'Sudadera de bota alta', '8', 'null', '32000'), ('302', '5', 'Conjunto corto', '8', 'null', '25000');

Ticket de Salida:



INSERT INTO `ticket_salida` (`IdTicketSalida`, `FechaSalida`, `PrecioTotal`) VALUES ('1', '2024-07-19', '50000'), ('2', '2024-08-19', '25000'), ('3', '2024-08-19', '100000');

Dtalle de Salida:



INSERT INTO `detalle_salida` (`ProductoCodigoProducto`, `Ticket SalidaIdTicketSalida`, `NombreProducto`, `Descricpción`, `Cantidad`, `ValorUnitario`) VALUES ('304', '1', 'Chaqueta impermeable acolchonada', NULL, '1', '50000'), ('306', '2', 'Pantaloneta corta', NULL, '1', '40000'), ('301', '3', 'Sudadera de hombre', NULL, '4', '25000');

Triggers:

Trigger de encriptar contraseñas

- Para insertar datos

DELIMITER //

CREATE TRIGGER before_insert_login
BEFORE INSERT ON login
FOR EACH ROW
BEGIN
SET NEW.password = AES_ENCRYPT(NEW.password, @encryption_key);
END;

DELIMITER;

//

- Para encriptar al actualizar los datos

DELIMITER //

CREATE TRIGGER before_update_login BEFORE UPDATE ON login

```
FOR EACH ROW
BEGIN
 SET NEW.password = AES_ENCRYPT(NEW.password, @encryption_key);
//
DELIMITER;
Sintaxis para desencriptar las contraseñas:
SELECT *, AES_DECRYPT(password, 'LANJ2024') AS password1 FROM login;
Procedimientos:
   - Mostrar nombre de la categoria de cada producto
BEGIN
 SELECT
   p.CodigoProducto,
   p.Nombre AS Nombre Producto,
   c.Nombre AS Nombre Categoria
 FROM
   producto p
 LEFT JOIN
   categoria c ON p.CategoriaCodigoCategoría = c.CodigoCategoría;
```

END

CALL `Mostrar nombre de la categoria de cada producto`();

Resultados de la ejecución de la rutina 'Mostrar nombre de la categoria de cada producto'

CodigoProducto	NombreProducto	NombreCategoria
301	Sudadera de Hombre	Sudaderas
302	Conjuto de Sudadera	Conjuntos
303	Sudadera de dama	Sudaderas
304	chaqueta impermeable	Chaquetas
305	Sudadera de Dama	Sudaderas
306	Pantaloneta corta	Pantalonetas

- Mostrar usuario INACTIVOS

SELECT

u.Num_Documento,

u.Nombres,

e.tipoEstado AS Estado

FROM

usuario u

JOIN

estado e ON u.EstadoCodigoEstado = e.CodigoEstado

WHERE

e.tipoEstado = 'INACTIVO'

Resultados de la ejecución de la rutina 'MostrarUsuariosInactivos'

Num_DocumentoNombresEstado1004785525Angie DanielaINACTIVO

- Mostrar usuarios que solo sean empleados

```
SELECT

u.Num_Documento,

u.Nombres,

r.tipoRol AS rol

FROM

usuario u

JOIN

rol r ON u.RolidRol= r.idRol

WHERE

r.tipoRol = 'Empleado'

Resultados de la ejecución de la rutina 'ObtenerUsuariosEmpleado'

Num_Documento Nombres rol

80461462 Jonny Duverney Empleado
```

- Crear un nuevo usuario

Kevin Santiago

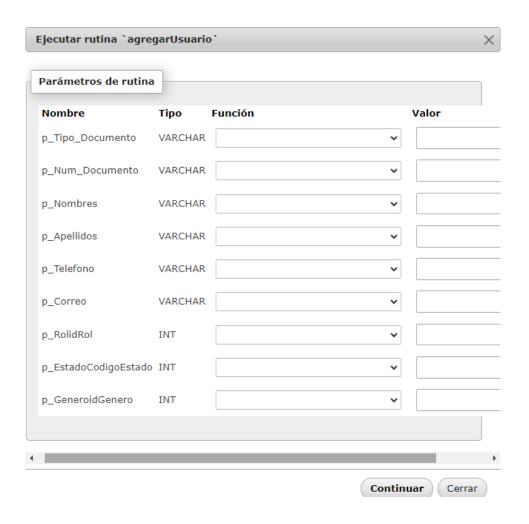
BEGIN

1033743193

INSERT INTO usuario (Tipo_Documento, Num_Documento, Nombres, Apellidos, Telefono, Correo, RolidRol, EstadoCodigoEstado, GeneroidGenero)

Empleado

VALUES (p_Tipo_Documento, p_Num_Documento, p_Nombres, p_Apellidos, p_Telefono, p_Correo, p_RolidRol, p_EstadoCodigoEstado, p_GeneroidGenero); END



Funciones:

- Total de productos que hay en la tabla

BEGIN

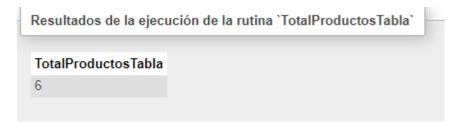
DECLARE total INT;

SELECT COUNT(*) INTO total

FROM producto;

RETURN total;

END



CONSULTAS

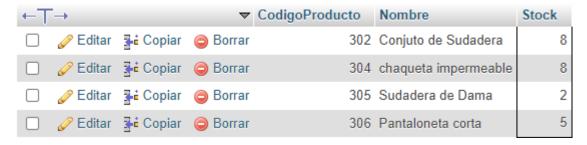
• Obtener el listado de los productos ordenados por volumen de precio mostrando los de menores precio.



CODIGO:

SELECT CodigoProducto, Nombre, Precio FROM producto ORDER BY Precio ASC;

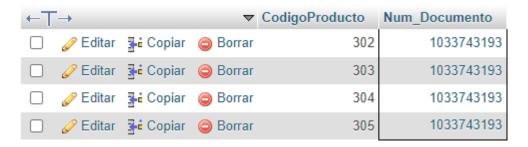
• Obtener el listado de los productos con stock menor a 12



CODIGO:

SELECT CodigoProducto, Nombre, Stock FROM producto WHERE Stock < 12;

• Listar todos los productos que no tengan los numero 8 y 2 en el número de documento del proveedor.



CODIGO:

SELECT CodigoProducto, Num_Documento FROM producto WHERE Num_Documento NOT LIKE '%8%' AND Num_Documento NOT LIKE '%2%';

Selecciona productos que contengan en su categoría el numero "5"



CODIGO:

SELECT CodigoProducto, Nombre, CategoriaCodigoCategoría FROM producto WHERE CategoriaCodigoCategoría LIKE '%5%';

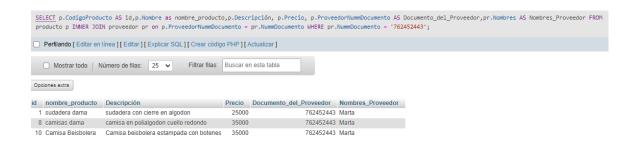
 Listar los usuarios del rol 21 indicando para cada uno el nombre de su tipo de rol.

Nombres	RolidRol	tipoRol
Jonny Duverney	21	Empleado
Kevin Santiago	21	Empleado

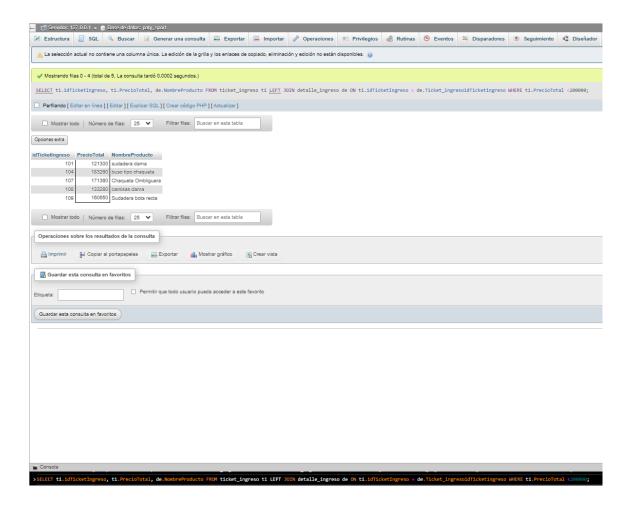
CODIGO:

SELECT u.Nombres, u.RolidRol, r.tipoRol FROM usuario u LEFT JOIN rol r ON u.RolidRol = r.idRol WHERE r.idRol = 21;

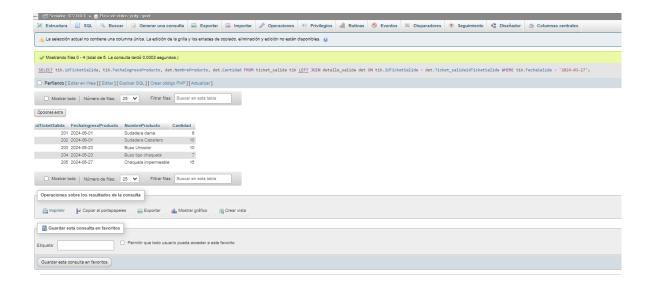
• Mostrar los productos de un proveedor específico buscado con su documento cambiando los nombres principales de las columnas colocando alias.



 Hacer la consulta donde me traiga el id del ticket del ingreso de un producto, donde muestre el precio total del ingreso menores a 200.000 y que de la tabla detalle producto me muestre solo los nombres de los productos con la condición propuesta en precio total.



 Hacer la consulta donde me muestre el id del ticket de salida de un producto, donde se muestre la fecha de Saida donde la condición sea que la fecha de salida se mayor 2024-03-27, igual que muestre el nombre y la cantidad del producto que salió, encontrado en el detalle de la salida.



Consola

>SELECT tik idTicketSalida, tik FechaIngresoProducto, det KombreProducto, det Cantidad FRON ticket_salida tik LEFT NON detalle_salida det ON tik IdTicketSalida = det Ticket_salidafdicketSalida WERE tik FechaSalida > '2024-83-27';