



# Proyecto Final - Bases de Datos

Antonio Javier Álvarez  
García

---



# Proyecto Final Bases de Datos

## Enunciado.

Vamos a proceder a crear la base de datos para la tienda MEDACJAVI, dicha tienda, va a estar compuesta por PRODUCTOS, CLIENTES, sus queridos EMPLEADOS y por supuesto, los PEDIDOS que realizan sus clientes en ella.

Los productos, tendrán un IDProducto, su correspondiente código de barras, el tipo de producto, el color del producto, el tamaño, su forma, su distribuidor, las unidades vendidas y la nacionalidad de origen.

En cuanto a los clientes, es importante saber que tendrán un IDCliente, su respectivo nombre, sus respectivos apellidos, su teléfono, su DNI, su producto favorito, su empleado favorito, su número de pedidos realizados, su fecha de nacimiento, su nacionalidad y su historial con el dinero que han gastado.

En cuanto a los empleados, es importante conocer su IDEmpleado, su número de ventas, su nombre, sus apellidos, su teléfono, su DNI, su nacionalidad, sus horas trabajadas, su sueldo mensual, su sueldo anual y su fecha de nacimiento.

En cuanto a los pedidos, es importante conocer el IDPedido, su distribuidor, su lugar de origen, su peso, su tamaño, el material de la caja, el precio del pedido y el tipo de envío.

Además, es importante conocer el STOCK que nos queda en la tienda, así como las direcciones de los clientes, que estarán compuestas de Calle, número, código postal, ciudad y provincia.



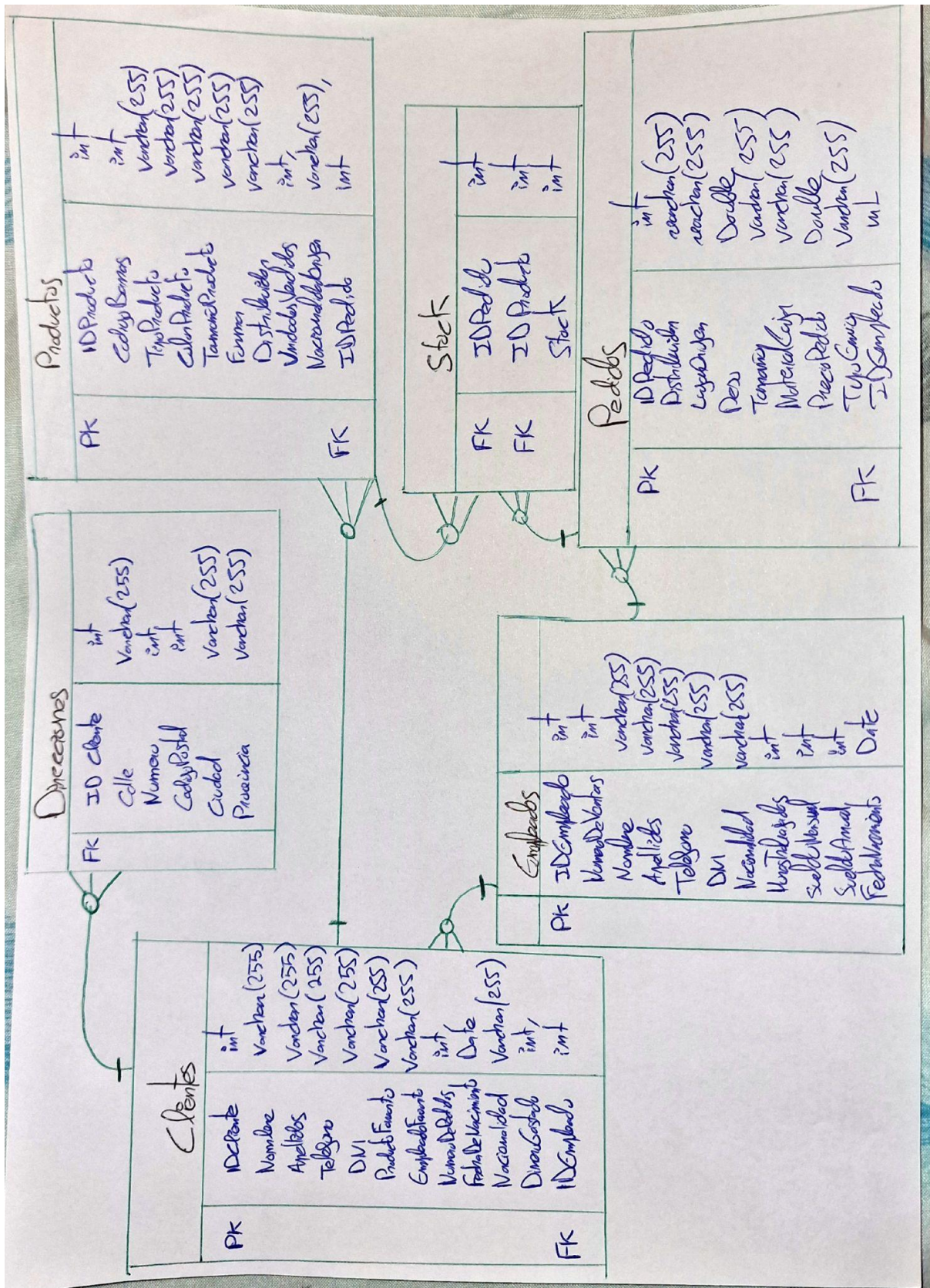
Es importante conocer, que un cliente, siempre será atendido por un único empleado, ya que queremos que sea atendido lo mejor posible, pero dicho empleado podrá atender a uno o muchos clientes. Es importante conocer también, que los clientes tienen una o muchas direcciones.

Los empleados, pueden realizar uno o muchos pedidos, pues es su trabajo.

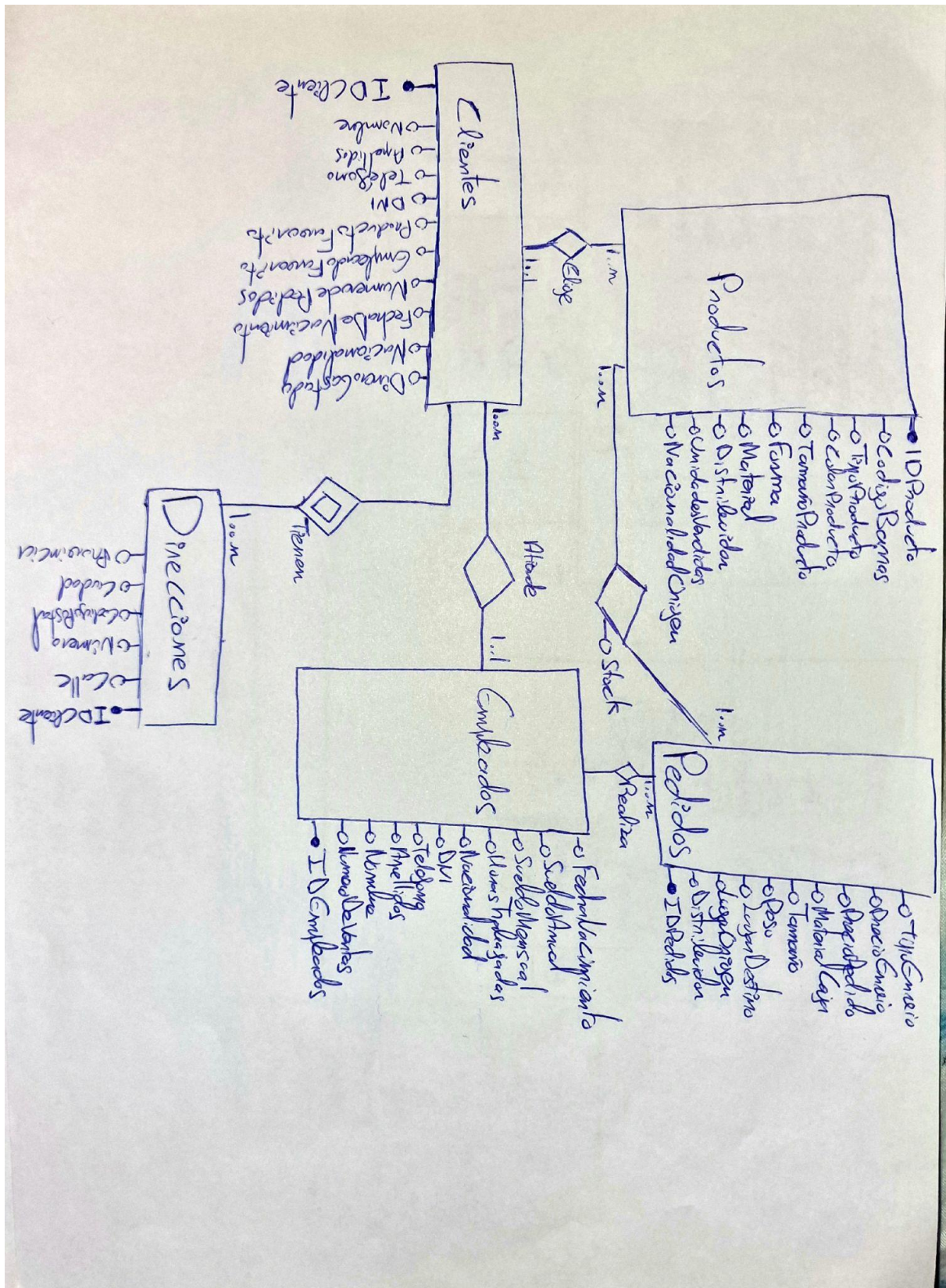
Los pedidos, están compuestos por uno o muchos productos, y los productos pueden estar presentes en uno o varios pedidos, por lo que es importante conocer su stock.

### **Modelo E-R, modelo relacional y paso a tablas.**













Productos (ID Producto, Código Barras, Tipo Producto, Color Producto, Tamaño Producto, Forma, material, Distribuidor, Unidades Vendidas, Nacionalidad Origen, ID Pedido)

Pedidos (ID Pedido, Distribuidor, Lugar Origen, Lugar Destino, Peso, Tamaño, Material Caja, Precio Pedido, Tipo Envase, ID Empleado)

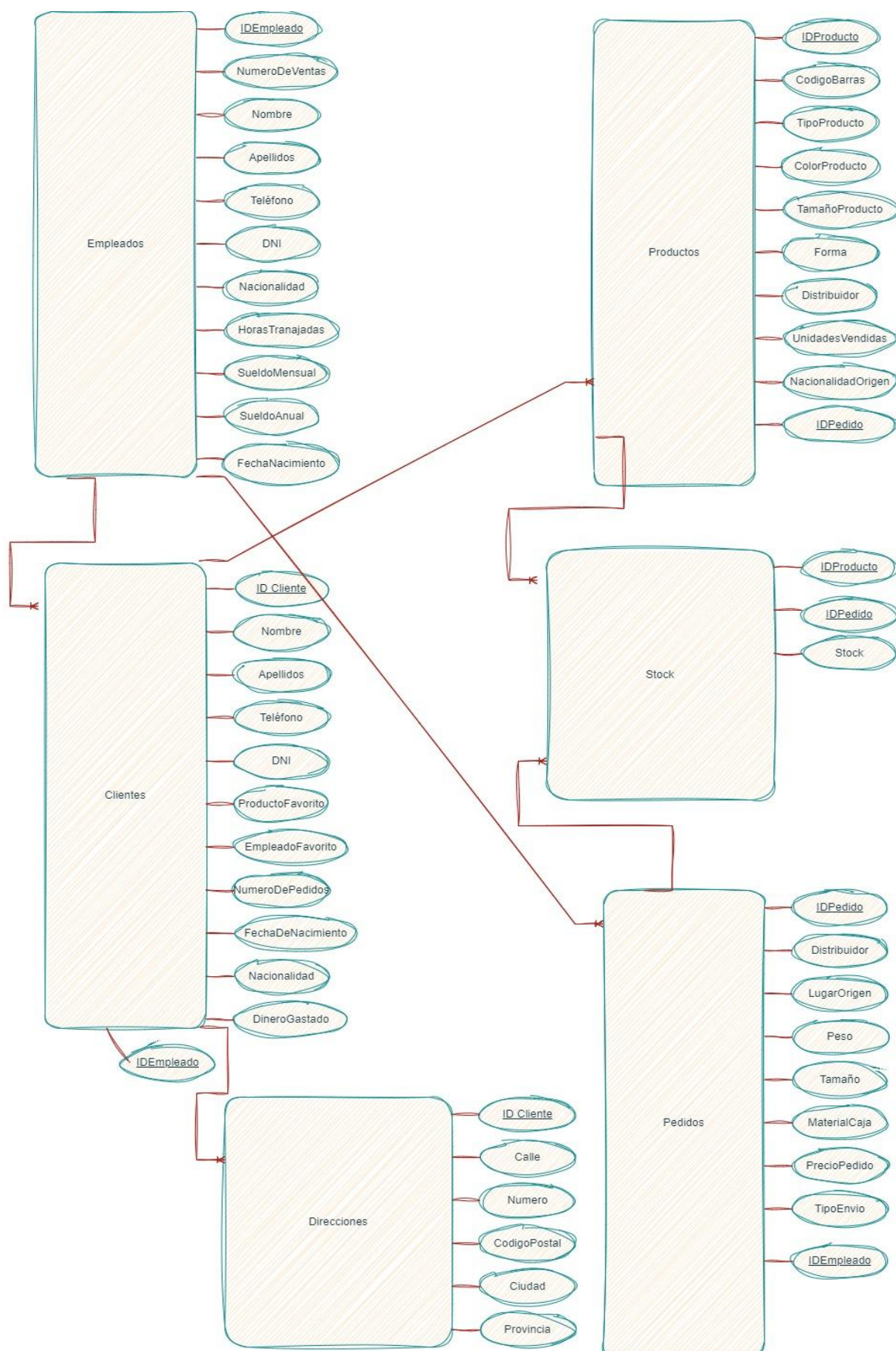
Empleados (ID Empleado, Numero De Ventas, Nombre, Apellidos, Teléfono, DNI, Nacionalidad, Horas Trabajadas, Sueldo Mensual, Sueldo Anual, Fecha Nacimiento)

Clientes (ID Cliente, ID Empleado, Nombre, Apellidos, Teléfono, DNI, Producto Favorito, Empleado Favorito, Numero De Pedidos, Fecha De Nacimiento, Nacionalidad, Dinero Gastado)

Direcciones (ID Cliente, Calle, Numero, Código Postal, Ciudad, Provincia)

Stock (ID Pedido, ID Producto, Stock)







Creación de la tabla y Base de Datos.

```
CREATE DATABASE ProyectoFinalJavierAlvarez;
```

```
USE ProyectoFinalJavierAlvarez;
```

```
/* Creamos las tablas de la base de datos */
```

```
CREATE TABLE Empleados (  
    IDEmpleado int NOT NULL PRIMARY KEY,  
    NumeroDeVentas int,  
    Nombre varchar(255),  
    Apellidos varchar(255),  
    Telefono varchar(255),  
    DNI varchar(255),  
    Nacionalidad varchar(255),  
    HorasTrabajadas int,  
    SueldoMensual int,  
    SueldoAnual int,  
    FechaNacimiento date
```





);

```
CREATE TABLE Productos(  
IDProducto int NOT NULL PRIMARY KEY,  
  
CodigoBarras int NOT NULL,  
  
TipoProducto varchar(255),  
  
ColorProducto varchar(255),  
  
TamañoProducto varchar(255),  
  
Forma varchar(255),  
  
Distribuidor varchar(255),  
  
UnidadesVenidas int,  
  
NacionalidadOrigen varchar(255),  
  
IDPedido int  
  
);
```

```
CREATE TABLE Pedidos (  
  
IDPedido int NOT NULL PRIMARY KEY,  
  
Distribuidor varchar(255),
```



LugarOrigen varchar(255),

Peso double,

Tamaño varchar(255),

MaterialCaja varchar(255),

PrecioPedido double,

TipoEnvio varchar(255),

IDEmpleado int

);

CREATE TABLE Clientes (

IDCliente int NOT NULL PRIMARY KEY,

Nombre varchar(255),

Apellidos varchar(255),

Telefono varchar(255),

DNI varchar(255),

ProductoFavorito varchar(255),

EmpleadoFavorito varchar(255),

NumeroDePedidos int,





```
FechaDeNacimiento date,  
  
Nacionalidad varchar(255),  
  
DineroGastado int,  
  
IDEmpleado int  
  
);
```

```
CREATE TABLE Direcciones(  
  
Calle varchar(255),  
  
Numero int,  
  
CodigoPostal int,  
  
Ciudad varchar(255),  
  
Provincia varchar(255),  
  
IDCliente int  
  
);
```

```
CREATE TABLE Stock(  
  
Stock int,
```



```
IDPedido int,
```

```
IDProducto int
```

```
);
```

*/\* Ahora vamos a establecer las FOREIGN KEYS \*/*

```
ALTER TABLE Stock
```

```
ADD FOREIGN KEY (IDPedido) REFERENCES Pedidos(IDPedido);
```

```
ALTER TABLE Stock
```

```
ADD FOREIGN KEY (IDProducto) REFERENCES Productos(IDProducto);
```

```
ALTER TABLE Clientes
```

```
ADD FOREIGN KEY (IDEmpleado) REFERENCES Empleados(IDEmpleado);
```

```
ALTER TABLE Pedidos
```

```
ADD FOREIGN KEY (IDEmpleado) REFERENCES Empleados(IDEmpleado);
```

```
ALTER TABLE Productos
```





```
ADD FOREIGN KEY (IDPedido) REFERENCES Pedidos(IDPedido);
```

```
ALTER TABLE Direcciones
```

```
ADD FOREIGN KEY (IDCliente) REFERENCES Clientes(IDCliente);
```

```
USE proyectofinaljavieralvarez;
```

```
INSERT INTO Empleados (IDEmpleado, NumeroDeVentas, Nombre, Apellidos,  
Telefono, DNI, Nacionalidad, HorasTrabajadas, SueldoMensual, SueldoAnual,  
FechaNacimiento)
```

```
VALUES
```

```
(1, 10, 'Juan', 'Perez', '111-222-333', '11111111A', 'Española', 160, 2000, 24000,  
'1980-01-01'),
```

```
(2, 5, 'Maria', 'Garcia', '444-555-666', '22222222B', 'Española', 120, 1800, 21600,  
'1985-02-02'),
```

```
(3, 15, 'Pedro', 'Lopez', '777-888-999', '33333333C', 'Española', 180, 2200, 26400,  
'1975-03-03'),
```

```
(4, 12, 'Sara', 'Fernandez', '000-111-222', '44444444D', 'Española', 150, 1900,  
22800, '1990-04-04'),
```



```
(5, 8, 'Javier', 'Martinez', '333-444-555', '55555555E', 'Española', 130, 1700,
20400, '1988-05-05'),

(6, 18, 'Ana', 'Rodriguez', '666-777-888', '66666666F', 'Española', 200, 2400,
28800, '1983-06-06'),

(7, 6, 'Manuel', 'Sanchez', '999-000-111', '77777777G', 'Española', 110, 1600,
19200, '1986-07-07'),

(8, 20, 'Lucia', 'Gomez', '222-333-444', '88888888H', 'Española', 220, 2600, 31200,
'1977-08-08'),

(9, 9, 'Raul', 'Pascual', '555-666-777', '99999999I', 'Española', 140, 1800, 21600,
'1984-09-09'),

(10, 14, 'Marta', 'Santos', '888-999-000', '11111111J', 'Española', 170, 2100,
25200, '1978-10-10'),

(11, 11, 'David', 'Jimenez', '111-222-333', '22222222K', 'Española', 160, 2000,
24000, '1982-11-11'),

(12, 7, 'Carmen', 'Castro', '444-555-666', '33333333L', 'Española', 120, 1600,
19200, '1989-12-12');
```

```
INSERT INTO Pedidos (IDPedido, Distribuidor, LugarOrigen, Peso, Tamaño,
MaterialCaja, PrecioPedido, TipoEnvio, IDEmpleado)
```

```
VALUES
```

```
(1, 'Distribuidor A', 'Madrid', 10.5, 'Grande', 'Cartón', 100.50, 'Urgente', 1),
(2, 'Distribuidor A', 'Barcelona', 8.2, 'Pequeño', 'Plástico', 50.00, 'Normal', 2),
```





(3, 'Distribuidor B', 'Valencia', 5.0, 'Mediano', 'Madera', 75.25, 'Urgente', 3),  
(4, 'Distribuidor C', 'Sevilla', 12.3, 'Grande', 'Cartón', 90.00, 'Urgente', 4),  
(5, 'Distribuidor B', 'Madrid', 4.7, 'Pequeño', 'Plástico', 30.50, 'Normal', 5),  
(6, 'Distribuidor A', 'Barcelona', 7.8, 'Mediano', 'Cartón', 65.75, 'Urgente', 6),  
(7, 'Distribuidor B', 'Valencia', 11.2, 'Grande', 'Madera', 120.00, 'Urgente', 7),  
(8, 'Distribuidor C', 'Sevilla', 2.5, 'Pequeño', 'Plástico', 20.00, 'Normal', 8),  
(9, 'Distribuidor A', 'Madrid', 15.0, 'Grande', 'Cartón', 200.50, 'Urgente', 9),  
(10, 'Distribuidor C', 'Barcelona', 9.6, 'Mediano', 'Cartón', 85.25, 'Urgente', 10),  
(11, 'Distribuidor B', 'Valencia', 6.4, 'Pequeño', 'Plástico', 40.00, 'Normal', 11),  
(12, 'Distribuidor C', 'Sevilla', 3.2, 'Mediano', 'Madera', 55.75, 'Urgente', 12);

INSERT INTO Productos (IDProducto, CodigoBarras, TipoProducto,  
ColorProducto, TamañoProducto, Forma, Distribuidor, UnidadesVenidas,  
NacionalidadOrigen, IDPedido) VALUES

(1, 123456789, 'Electrónico', 'Negro', 'Grande', 'Rectangular', 'Samsung', 100,  
'Corea del Sur', 1),  
(2, 987654321, 'Electrónico', 'Blanco', 'Pequeño', 'Circular', 'Apple', 50, 'Estados  
Unidos', 2),  
(3, 567891234, 'Mueble', 'Marrón', 'Mediano', 'Cuadrado', 'Ikea', 200, 'Suecia', 3),



(4, 234567891, 'Ropa', 'Azul', 'Pequeño', 'Redondo', 'Zara', 150, 'España', 4),  
(5, 789123456, 'Electrónico', 'Plata', 'Mediano', 'Triangular', 'LG', 75, 'Corea del Sur', 5),  
(6, 345678912, 'Herramienta', 'Negro', 'Grande', 'Rectangular', 'Bosch', 50, 'Alemania', 6),  
(7, 891234567, 'Juguete', 'Rojo', 'Pequeño', 'Circular', 'Hasbro', 100, 'Estados Unidos', 7),  
(8, 456789123, 'Electrodoméstico', 'Gris', 'Grande', 'Cuadrado', 'Samsung', 25, 'Corea del Sur', 8),  
(9, 912345678, 'Libro', 'Verde', 'Mediano', 'Rectangular', 'Penguin Random House', 300, 'Reino Unido', 9),  
(10, 678912345, 'Joyería', 'Oro', 'Pequeño', 'Redondo', 'Cartier', 10, 'Francia', 10),  
(11, 234567890, 'Ropa', 'Negro', 'Grande', 'Cuadrado', 'Adidas', 125, 'Alemania', 11),  
(12, 890123456, 'Herramienta', 'Azul', 'Mediano', 'Triangular', 'Makita', 75, 'Japón', 12);

INSERT INTO Clientes (IDCliente, Nombre, Apellidos, Telefono, DNI, ProductoFavorito, EmpleadoFavorito, NumeroDePedidos, FechaDeNacimiento, Nacionalidad, DineroGastado, IDEmpleado) VALUES





- (1, 'Juan', 'García', '123456789', '12345678A', 'Camisa Roja', 'María Sánchez', 5, '1980-01-01', 'Español', 1000, 1),
- (2, 'Ana', 'Martínez', '987654321', '87654321B', 'Pantalón Azul', 'Pedro Gómez', 2, '1995-05-10', 'Mexicano', 500, 2),
- (3, 'Pedro', 'González', '654321987', '65432198C', 'Camiseta Verde', 'María Sánchez', 3, '1990-12-24', 'Colombiano', 700, 3),
- (4, 'María', 'Sánchez', '147258369', '14725836D', 'Chaqueta Negra', 'Juan García', 4, '1988-06-15', 'Español', 800, 4),
- (5, 'Luisa', 'López', '258369147', '25836914E', 'Pantalón Rojo', 'Juan García', 6, '1985-09-30', 'Mexicano', 1200, 5),
- (6, 'Miguel', 'Fernández', '369147258', '36914725F', 'Camisa Blanca', 'Pedro Gómez', 1, '2000-03-05', 'Español', 200, 6),
- (7, 'Sara', 'Gómez', '963852741', '96385274G', 'Jersey Gris', 'Pedro Gómez', 2, '1998-07-20', 'Mexicano', 400, 7),
- (8, 'David', 'Pérez', '321654987', '32165498H', 'Camisa Azul', 'María Sánchez', 3, '1993-11-02', 'Colombiano', 600, 8),
- (9, 'Laura', 'Rodríguez', '456789123', '45678912I', 'Vestido Negro', 'Juan García', 5, '1987-04-10', 'Español', 900, 9),
- (10, 'Pablo', 'Jiménez', '654789321', '65478932J', 'Pantalón Marrón', 'Pedro Gómez', 1, '1997-08-12', 'Mexicano', 200, 10),
- (11, 'Lucía', 'Hernández', '852963741', '85296374K', 'Jersey Morado', 'Juan García', 2, '1992-02-28', 'Español', 300, 11),



(12, 'Carlos', 'Sánchez', '987123456', '98712345L', 'Chaqueta Marrón', 'María  
Sánchez', 4, '1989-10-18', 'Colombiano', 1000, 12);

INSERT INTO Direcciones (Calle, Numero, CodigoPostal, Ciudad, Provincia,  
IDCliente) VALUES

('Calle Mayor', 1, 28001, 'Madrid', 'Madrid', 1),

('Calle del Sol', 2, 41001, 'Sevilla', 'Sevilla', 1),

('Calle de la Luna', 3, 50001, 'Zaragoza', 'Zaragoza', 2),

('Calle de la Estrella', 4, 03001, 'Alicante', 'Alicante', 2),

('Calle del Mar', 5, 08001, 'Barcelona', 'Barcelona', 3),

('Calle de la Montaña', 6, 29001, 'Málaga', 'Málaga', 3),

('Calle de la Playa', 7, 35001, 'Las Palmas', 'Las Palmas', 4),

('Calle del Puerto', 8, 46001, 'Valencia', 'Valencia', 4),

('Calle de la Ría', 9, 48001, 'Bilbao', 'Vizcaya', 5),

('Calle del Arenal', 10, 41001, 'Sevilla', 'Sevilla', 5),

('Calle del Jardín', 11, 14001, 'Córdoba', 'Córdoba', 6),

('Calle del Río', 12, 46001, 'Valencia', 'Valencia', 6),

('Calle del Bosque', 13, 33001, 'Oviedo', 'Asturias', 7),

('Calle de la Fuente', 14, 18001, 'Granada', 'Granada', 7),

('Calle de la Plaza', 15, 30001, 'Murcia', 'Murcia', 8),





('Calle del Parque', 16, 11001, 'Cádiz', 'Cádiz', 8),

('Calle de la Torre', 17, 35001, 'Las Palmas', 'Las Palmas', 9),

('Calle del Paseo', 18, 46001, 'Valencia', 'Valencia', 9),

('Calle del Teatro', 19, 02001, 'Albacete', 'Albacete', 10),

('Calle del Palacio', 20, 26001, 'Logroño', 'La Rioja', 10);

INSERT INTO Stock (Stock, IDPedido, IDProducto) VALUES

(50, 1, 1),

(20, 2, 2),

(100, 3, 3),

(75, 4, 4),

(30, 5, 5),

(50, 6, 6),

(80, 7, 7),

(15, 8, 8),

(120, 9, 9),

(40, 10, 10),

(60, 11, 11),

(25, 12, 12);



Creación de consultas.

-- Te devuelve los 5 Clientes y Empleados más jóvenes

```
SELECT Clientes.Nombre AS NombreCliente, Clientes.Apellidos AS
ApellidosCliente, Clientes.FechaDeNacimiento AS FechaNacimientoCliente,

Empleados.Nombre AS NombreEmpleado, Empleados.Apellidos AS
ApellidosEmpleado, Empleados.FechaNacimiento AS FechaNacimientoEmpleado

FROM CLIENTES

INNER JOIN Empleados ON Clientes.IDEmpleado = Empleados.IDEmpleado

ORDER BY FechaNacimientoCliente, FechaNacimientoEmpleado DESC

LIMIT 5;
```

-- Consulta los tres empleados que más han trabajado

```
Select *

From Empleados

ORDER BY HorasTrabajadas DESC

LIMIT 3;
```



-- Muestra cuantos envíos de cada tipo se han realizado

```
SELECT Count(Pedidos.TipoEnvio) as NumeroDeEnvios,Pedidos.TipoEnvio  
  
FROM Pedidos  
  
INNER JOIN Productos  
  
ON Pedidos.IDPedido = Productos.IDPedido  
  
GROUP BY Pedidos.TipoEnvio;
```

-- Mostará el número de ventas de cada empleado, el nombre y el distribuidor

```
SELECT Empleados.NumeroDeVentas, Empleados.Nombre,  
        Productos.Distribuidor  
  
FROM Empleados INNER JOIN Pedidos  
  
ON Empleados.IDEmpleado = Pedidos.IDEmpleado  
  
INNER JOIN Productos  
  
ON Pedidos.IDPedido = Productos.IDPedido
```





GROUP BY Empleados.NumeroDeVentas, Empleados.Nombre,  
Productos.Distribuidor

ORDER BY Productos.Distribuidor ASC;

-- Consulta los tres empleados con más sueldo anual

Select \*

From Empleados

ORDER BY SueldoAnual DESC

LIMIT 3;

-- Te devuelve las direcciones de cada cliente

-- Usamos producto cartesiano

SELECT Clientes.Nombre, Clientes.Apellidos, Direcciones.Calle,  
Direcciones.Numero, Direcciones.CodigoPostal, Direcciones.Ciudad

FROM Clientes, Direcciones

WHERE Clientes.IDCliente = Direcciones.IDCliente;



-- TOP Clientes que más dinero se han gastado

SELECT \*

FROM Clientes

ORDER BY DineroGastado

LIMIT 3;

-- Tipo de Envío realizado por cada empleado por orden alfabético

SELECT Pedidos.TipoEnvio, Empleados.Nombre

FROM Empleados INNER JOIN Pedidos

ON Empleados.IDEmpleado = Pedidos.IDEmpleado

INNER JOIN Productos

ON Pedidos.IDPedido = Productos.IDPedido

ORDER BY Empleados.Nombre;

-- Selecciona los tres productos con más stock

Select \*

From Stock



order by stock DESC

LIMIT 3;

-- Los tres productos más vendidos

SELECT Productos.TipoProducto, Productos.UnidadesVenidas

FROM Pedidos

INNER JOIN Productos ON Pedidos.IDPedido = Productos.IDPedido

GROUP BY Productos.TipoProducto, Productos.UnidadesVenidas

ORDER BY Productos.UnidadesVenidas DESC

LIMIT 3;

-- Consulta todas las horas trabajadas por los empleados

SELECT Sum(HorasTrabajadas)

FROM Empleados;

-- Cantidad total de productos vendidos, Peso total de los paquetes enviados y  
número de empleados de la empresa





```
SELECT sum(Productos.UnidadesVenidas) as NdeVentas, sum(Pedidos.Peso) as  
PesoTotal, count(Empleados.IDEmpleado) as TotalEmpeados
```

```
FROM Empleados
```

```
INNER JOIN Pedidos
```

```
ON Empleados.IDEmpleado = Pedidos.IDEmpleado
```

```
INNER JOIN Productos
```

```
ON Pedidos.IDPedido = Productos.IDPedido;
```

-- TOP 5 Clientes que más media de gasto tienen, ventas medias por empleado y  
ventas medias entre todos los empleados y nacionalidad de origen de los  
productos

```
SELECT AVG(Clientes.DineroGastado) as MediaGastoClientes,  
AVG(Empleados.NumeroDeVentas) as MediaVentasEmpleados,  
Productos.NacionalidadOrigen
```

```
FROM Clientes
```

```
INNER JOIN Empleados
```

```
ON Clientes.IDEmpleado = Empleados.IDEmpleado
```

```
INNER JOIN Pedidos
```

```
ON Empleados.IDEmpleado = Pedidos.IDEmpleado
```

```
INNER JOIN Productos
```

```
ON Pedidos.IDPedido = Productos.IDPedido
```



GROUP BY Productos.NacionalidadOrigen

ORDER BY MediaGastoClientes DESC

LIMIT 5;

### Creación de Triggers.

-- Este trigger actualiza el precio total del pedido cuando insertas más pedidos

DELIMITER // -- En MySQLWorkbench es necesario utilizarlo para que funcionen los triggers, ya que ; puede originar problemas al utilizarse

CREATE TRIGGER actualizar\_precio\_pedido

AFTER INSERT ON Stock

FOR EACH ROW

BEGIN

UPDATE Pedidos

SET PrecioPedido = (SELECT SUM(Productos.PrecioProducto \* Stock.Stock)

FROM Productos

INNER JOIN Stock ON Productos.IDProducto = Stock.IDProducto

WHERE Stock.IDPedido = NEW.IDPedido)

WHERE IDPedido = NEW.IDPedido;



END//

DELIMITER ;

-- Este trigger actualiza el número de pedidos realizados por un cliente cuando se inserta una nueva fila en pedidos

DELIMITER //

CREATE TRIGGER actualizar\_numero\_pedidos

AFTER INSERT ON Pedidos

FOR EACH ROW

BEGIN

UPDATE Clientes

SET NumeroDePedidos = NumeroDePedidos + 1

WHERE IDCliente = (SELECT IDCliente FROM Pedidos WHERE IDPedido = NEW.IDPedido);

END//

DELIMITER ;

-- Este trigger actualiza el SOTCK al insertar una nueva fila en la tabla



DELIMITER //

CREATE TRIGGER actualizar\_stock\_producto

AFTER INSERT ON Stock

FOR EACH ROW

BEGIN

    UPDATE Productos

    SET Stock = Stock - NEW.Stock

    WHERE IDProducto = NEW.IDProducto;

END//

DELIMITER ;

-- Cuando se elimina una fila de la tabla stock se actualiza el producto correspondiente

DELIMITER //

CREATE TRIGGER actualizar\_stock\_eliminar

AFTER DELETE ON Stock

FOR EACH ROW





BEGIN

UPDATE Productos

SET Stock = Stock + OLD.Stock

WHERE IDProducto = OLD.IDProducto;

END//

DELIMITER ;