



Base de datos con SQL SERVER

Proyecto Final:

Gestión integral de un E-Commerce

Autores:

Auccapure Galindo, Maritza

Hende Cisneros, Janice

Silva Arenas, Diana Lucía

Tume Díaz, Andrés

Docente:

Ing. Kevin Rivera Vergaray

Lima, Perú

2024

I. EXPLICACIÓN GENERAL DEL TRABAJO

Título del trabajo:

Gestión integral de un E-Commerce

Objetivo:

El objetivo principal de este trabajo es **gestionar y optimizar las operaciones de un E-Commerce utilizando SQL Server**, asegurando la integridad de los datos y la eficiencia en el manejo de la información. Este sistema debe registrar y consultar información relacionada con **clientes, productos, pedidos, pagos, envíos y tiendas/almacenes**. A través de consultas SQL, se busca optimizar la gestión del inventario, los pagos, el seguimiento de pedidos y el despacho de productos, con el fin de mejorar tanto la operatividad interna del negocio como la experiencia del cliente, así como también extraer información relevante para un análisis enfocado en una mejor toma de decisiones.

Entidades identificadas:

Para gestionar correctamente las operaciones del E-Commerce, se han identificado las siguientes **entidades clave** que componen el sistema de base de datos:

- **Clientes:** Representan a los usuarios que compran en la tienda. Cada cliente tiene información personal como nombre, tipo de documento, correo electrónico, dirección, teléfono, etc.
- **Productos:** Son los artículos que se venden en la tienda. Cada producto tiene atributos como nombre, descripción, precio, stock y categoría.
- **Pedidos:** Representan las compras realizadas por los clientes. Incluyen información sobre la tienda/almacén que gestiona el pedido, la fecha, el estado y el importe total.
- **Detalles del pedido:** Es el desglose de los productos adquiridos en un pedido, con detalles como cantidades y precios.
- **Tiendas:** Son los lugares físicos desde donde se gestionan los productos y los envíos. Cada tienda tiene un nombre, dirección, ciudad y departamento.
- **Métodos de pago:** Son las formas de pago utilizadas por los clientes para completar sus compras. Cada método de pago tiene un nombre (ej. tarjeta de crédito, PayPal, etc.).
- **Pagos:** Los pagos se registran al momento de realizar una compra. Incluyen información sobre el monto pagado, el método de pago utilizado, el estado del pago (procesado o rechazado) y la fecha de la transacción.
- **Envíos:** Una vez que un pedido es pagado y confirmado, se genera un envío que lleva el producto al cliente. Los datos del envío incluyen dirección, estado, fecha de envío y fecha de entrega.
- **Stock de tienda:** Registra la cantidad de productos disponibles en cada tienda/almacén, permitiendo gestionar el inventario en tiempo real.

Tablas y campos relevantes:

En este apartado, se enumerarán las tablas que representan las entidades mencionadas y los campos clave que almacenan la información esencial para la operación del E-Commerce.

Para efectos de este trabajo, se limitó el número de tablas/entidades a diez. Sin embargo, la información puede abarcar un número mayor. Por ejemplo, se omitieron tablas de proveedores y comentarios/reseñas.

Cientes

- **id:** Identificador único del cliente.
- **tipo_documento:** Tipo de documento de identificación (DNI, pasaporte, etc.).
- **numero_documento:** Número de documento único para cada cliente.
- **apellido_paterno, apellido_materno, nombre:** Nombres y apellidos del cliente.
- **email:** Correo electrónico único del cliente.
- **direccion, ciudad, departamento:** Dirección de envío.
- **telefono:** Número telefónico del cliente.
- **fecha_nacimiento:** Fecha de nacimiento del cliente (opcional).
- **fecha_registro:** Fecha en la que el cliente se registró en el sistema.

Productos

- **id:** Identificador único del producto.
- **nombre:** Nombre del producto.
- **descripcion:** Descripción detallada del producto.
- **precio:** Precio unitario del producto.
- **stock:** Cantidad disponible del producto en inventario.
- **categoria_id:** Referencia a la categoría del producto.

Pedidos

- **id:** Identificador único del pedido.
- **cliente_id:** Referencia al cliente que realizó el pedido.
- **tienda_id:** Referencia a la tienda que gestiona el pedido.
- **fecha_pedido:** Fecha y hora en que se realizó el pedido.
- **total:** Monto total del pedido.
- **estado:** Estado del pedido (Pendiente, Enviado, Entregado, Rechazado).

Pagos

- **id:** Identificador único del pago.
- **pedido_id:** Referencia al pedido asociado con el pago.
- **metodo_pago_id:** Referencia al método de pago utilizado.
- **importe:** Monto pagado por el cliente.
- **estado:** Estado del pago (En Proceso, Confirmado, Rechazado).
- **fecha_pago:** Fecha en que se realizó el pago.

Envíos

- **id:** Identificador único del envío.
- **pedido_id:** Referencia al pedido asociado con el envío.

- **direccion_envio, ciudad_envio, departamento:** Datos de la dirección de envío.
- **referencia:** Código de referencia del envío.
- **estado_envio:** Estado del envío (En Preparación, En Tránsito, Entregado, Rechazado).
- **fecha_envio:** Fecha de salida del producto.
- **fecha_entrega:** Fecha de entrega del producto al cliente.

Tiendas

- **id:** Identificador único de la tienda.
- **nombre:** Nombre de la tienda.
- **direccion, ciudad, departamento:** Dirección de la tienda.

Stock_Tienda

- **id:** Identificador único del registro de stock.
- **producto_id:** Referencia al producto en la tienda.
- **tienda_id:** Referencia a la tienda o almacén.
- **cantidad:** Cantidad de producto disponible en la tienda.

Desafíos del trabajo:

Algunos de los desafíos que podrían surgir al gestionar este sistema con SQL Server son:

- **Gestión de grandes volúmenes de datos:** A medida que el E-Commerce crezca, se manejarán grandes cantidades de datos (clientes, pedidos, productos, pagos, etc.). La eficiencia en las consultas y el manejo del rendimiento de la base de datos será un desafío constante.
- **Manejo de stock en múltiples tiendas:** Asegurar que el stock de productos se actualice correctamente en tiempo real para evitar la sobreventa de productos en diferentes tiendas/almacenes.
- **Integración con sistemas de pago y envío:** Es posible que se deban integrar sistemas externos para procesar pagos y gestionar el envío de productos. Mantener la consistencia y sincronización entre estos sistemas y la base de datos será complejo.
- **Actualización de estados de pedidos y envíos:** Mantener los estados de los pedidos, pagos y envíos actualizados correctamente en el sistema, evitando inconsistencias, especialmente cuando se maneja un flujo de datos tan dinámico.

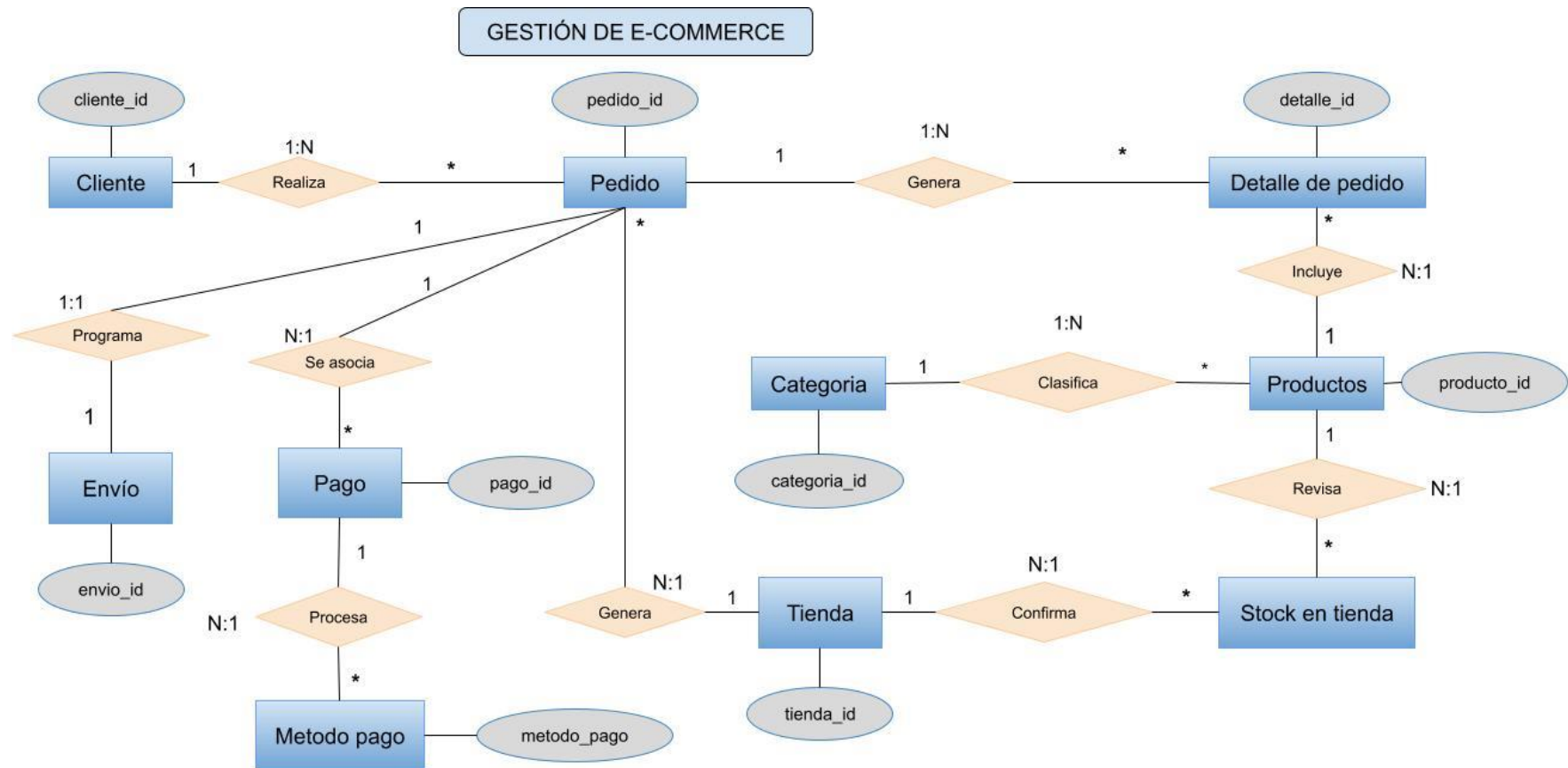
Beneficios esperados:

Al implementar este sistema en SQL Server, los **beneficios esperados** para la operación del E-Commerce son:

- **Optimización de la gestión del inventario:** La actualización automática de los stocks por tienda garantizará que la empresa no venda productos fuera de inventario, mejorando la experiencia del cliente.

- **Mejora en el análisis de ventas y comportamiento del cliente:** A través de consultas SQL, se podrán generar reportes detallados sobre ventas, productos más vendidos, pagos, etc., lo que permitirá a los responsables tomar decisiones informadas.
- **Mayor control sobre los pagos y envíos:** Al tener un sistema centralizado para registrar pagos y envíos, la empresa podrá gestionar de manera más eficiente el seguimiento de estos procesos, reduciendo errores.
- **Escalabilidad y fiabilidad:** SQL Server, siendo una base de datos robusta, asegura que el sistema será escalable y confiable, capaz de manejar un crecimiento en volumen de datos y usuarios sin afectar el rendimiento.

II. DISEÑO CONCEPTUAL DEL TRABAJO



III. CONSTRUCCIÓN DE TABLAS EN SQL SERVER

En base al diseño conceptual del trabajo y las entidades identificadas, se procedieron a crear las tablas en SQL Server Management Studio.

Creación de la base de datos en donde se realizará el trabajo:

```
CREATE DATABASE dsrp_ecommerce_g13;  
GO
```

```
USE dsrp_ecommerce_g13;  
GO
```

Creación de la tabla CLIENTES:

```
CREATE TABLE clientes(  
id INT PRIMARY KEY IDENTITY(1,1),  
tipo_documento VARCHAR(50) NOT NULL,  
numero_documento VARCHAR(50) UNIQUE NOT NULL,  
apellido_paterno VARCHAR(100) NOT NULL,  
apellido_materno VARCHAR(100) NOT NULL,  
nombre VARCHAR(100) NOT NULL,  
email VARCHAR(255) UNIQUE NOT NULL,  
direccion VARCHAR(255) NOT NULL,  
ciudad VARCHAR(100) NOT NULL,  
departamento VARCHAR(100) NOT NULL,  
telefono VARCHAR(15) NOT NULL,  
fecha_nacimiento DATE NULL,  
fecha_registro DATETIME DEFAULT GETDATE()  
);
```

Creación de la tabla CATEGORÍAS:

```
CREATE TABLE categorias (  
id INT PRIMARY KEY IDENTITY(1,1),  
nombre VARCHAR(255) UNIQUE NOT NULL,  
categoria_padre_id INT NULL,  
CONSTRAINT FK_categoria_padre_id FOREIGN KEY (categoria_padre_id) REFERENCES  
categorias(id)  
);
```

Creación de la tabla PRODUCTOS:

```
CREATE TABLE productos (  
id INT PRIMARY KEY IDENTITY(1,1),  
nombre VARCHAR(255) UNIQUE NOT NULL,  
descripcion TEXT,  
precio DECIMAL(10, 2) NOT NULL,  
stock INT NOT NULL,  
categoria_id INT NOT NULL,  
CONSTRAINT FK_categoria_id_productos FOREIGN KEY (categoria_id) REFERENCES  
categorias(id)  
);
```

Creación de la tabla TIENDAS:

```
CREATE TABLE tiendas (  
id INT PRIMARY KEY IDENTITY(1,1),  
nombre VARCHAR(255) UNIQUE NOT NULL,  
direccion VARCHAR(255) NOT NULL,  
ciudad VARCHAR(100) NOT NULL,  
departamento VARCHAR(100) NOT NULL  
);
```

Creación de la tabla PEDIDOS:

```
CREATE TABLE pedidos (  
id INT PRIMARY KEY IDENTITY(1,1),  
cliente_id INT NOT NULL,  
tienda_id INT NOT NULL,  
fecha_pedido DATETIME DEFAULT GETDATE() NOT NULL,  
total DECIMAL(10, 2) NOT NULL,  
estado VARCHAR(50) NOT NULL,  
CONSTRAINT FK_cliente_id_pedidos FOREIGN KEY (cliente_id) REFERENCES clientes(id),  
CONSTRAINT FK_tienda_id_pedidos FOREIGN KEY (tienda_id) REFERENCES tiendas(id)  
);
```

Creación de la tabla DETALLES DE PEDIDOS:

```
CREATE TABLE detalles_pedido (  
id INT PRIMARY KEY IDENTITY(1,1),  
pedido_id INT NOT NULL,  
producto_id INT NOT NULL,  
cantidad INT NOT NULL,  
precio DECIMAL(10, 2) NOT NULL,  
CONSTRAINT FK_pedido_id_detalle_pedido FOREIGN KEY (pedido_id) REFERENCES  
pedidos(id),  
CONSTRAINT FK_producto_id_detalle_pedido FOREIGN KEY (producto_id) REFERENCES  
productos(id)  
);
```

Creación de la tabla MÉTODOS DE PAGO:

```
CREATE TABLE metodo_pago (  
id INT PRIMARY KEY IDENTITY(1,1),  
nombre VARCHAR(50) NOT NULL  
);
```

Creación de la tabla PAGOS:

```
CREATE TABLE Pagos (  
id INT PRIMARY KEY IDENTITY(1,1),  
pedido_id INT NOT NULL,  
metodo_pago_id INT NOT NULL,  
importe DECIMAL(10, 2) NOT NULL,  
estado VARCHAR(50) NOT NULL,  
fecha_pago DATETIME DEFAULT GETDATE() NOT NULL,  
CONSTRAINT FK_pedido_id_pagos FOREIGN KEY (pedido_id) REFERENCES pedidos(id),  
CONSTRAINT FK_metodo_pago_id_pagos FOREIGN KEY (metodo_pago_id) REFERENCES  
metodo_pago(id)  
);
```


Creación de la tabla ENVÍOS:

```
CREATE TABLE envios(  
id INT PRIMARY KEY IDENTITY(1,1),  
pedido_id INT,  
direccion_envio VARCHAR(255) NOT NULL,  
ciudad_envio VARCHAR(100) NOT NULL,  
departamento VARCHAR(100) NOT NULL,  
referencia VARCHAR(100) NOT NULL,  
estado_envio VARCHAR(50) NOT NULL,  
fecha_envio DATETIME DEFAULT GETDATE(),  
fecha_entrega DATETIME,  
CONSTRAINT FK_pedido_id_envios FOREIGN KEY (pedido_id) REFERENCES pedidos(id)  
);
```

Creación de la tabla STOCK DE TIENDA:

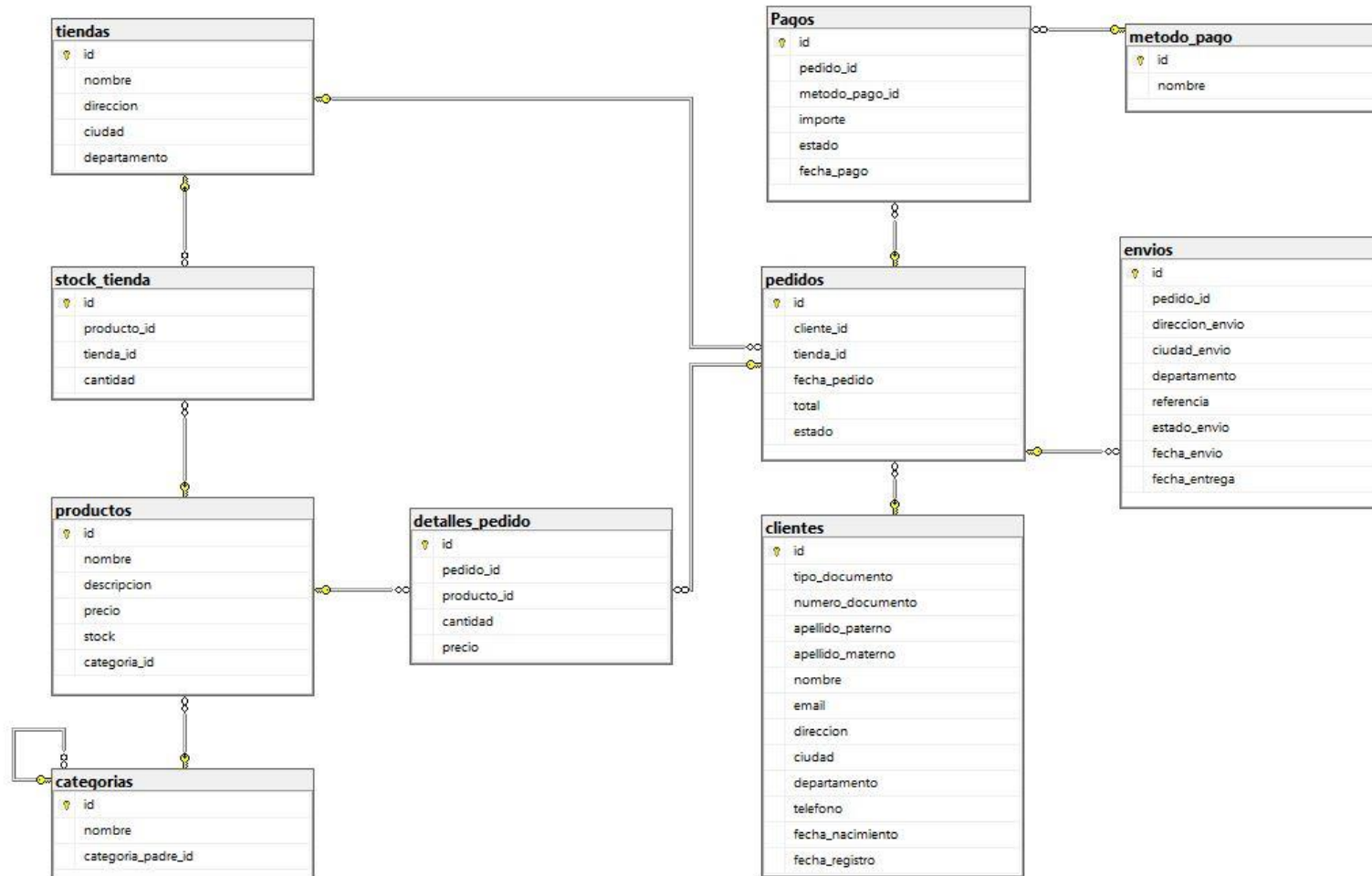
```
CREATE TABLE stock_tienda (  
id INT PRIMARY KEY IDENTITY(1,1),  
producto_id INT NOT NULL,  
tienda_id INT NOT NULL,  
cantidad INT NOT NULL,  
CONSTRAINT FK_producto_id_stock_tienda FOREIGN KEY (producto_id) REFERENCES  
productos(id),  
CONSTRAINT FK_tienda_id_stock_tienda FOREIGN KEY (tienda_id) REFERENCES tiendas(id)  
);
```

Modificación de tablas con campos cuyos valores será predeterminados:

```
--Estado del pedido: Pendiente, Enviado, Entregado, Rechazado  
ALTER TABLE pedidos  
ADD CONSTRAINT check_estado CHECK (estado IN  
( 'Pendiente', 'Enviado', 'Entregado', 'Rechazado' ));  
  
--Estado del pago realizado: En Proceso, Confirmado, Rechazado  
ALTER TABLE pagos  
ADD CONSTRAINT check_estado_pago CHECK (estado IN ( 'En  
Proceso', 'Confirmado', 'Rechazado' ));  
  
--Estado del envío del pedido: En Proceso, Confirmado, Rechazado  
ALTER TABLE envios  
  
ADD CONSTRAINT check_estado_envio CHECK (estado_envio IN ( 'En Preparación', 'En  
Tránsito', 'Entregado', 'Rechazado' ));
```

IV. DISEÑO FÍSICO DE TABLAS EN SQL SERVER

Una vez diseñadas las tablas, se generó el modelo físico utilizando SQL Server, para plasmar el flujo de las transacciones:



V. GENERACIÓN DE REGISTROS PARA CADA TABLA

Ya habiendo plasmado la figura general del flujo de trabajo, se procedió a generar información para cada tabla/entidad. Por la longitud del código para ciertas tablas, en algunos casos se dejarán los primeros registros como modelo.

USE dsrp_ecommerce_g13

GO

Registros para la tabla CLIENTES:

```
INSERT INTO Clientes (tipo_documento, numero_documento, apellido_paterno,
apellido_materno, nombre, email, direccion, ciudad, departamento, telefono,
fecha_nacimiento, fecha_registro)
VALUES
('DNI', '12345678', 'Pérez', 'González', 'Juan', 'juan.perez@email.com', 'Av. Siempre
Viva 123', 'Lima', 'Lima', '987654321', '1990-05-15', '2024-01-15 08:30:00'),
('CE', '20123456789012345679', 'Ramírez', 'Suárez', 'Carlos',
'carlos.ramirez@email.com', 'Calle Falsa 456', 'Arequipa', 'Arequipa', '987654322',
'1985-08-20', '2024-02-02 14:45:00'),
('DNI', '87654322', 'Lopez', 'Martínez', 'Ana', 'ana.lopez@email.com', 'Jr. Los Olivos
789', 'Cusco', 'Cusco', '987654323', '1992-02-10', '2024-03-05 11:15:00'),
('DNI', '23456790', 'Fernández', 'Alvarado', 'Luis', 'luis.fernandez@email.com', 'Av.
Las Palmas 101', 'Trujillo', 'La Libertad', '987654324', '1993-04-25', '2024-04-10
17:00:00'),
('CE', '20123456789012345680', 'Gómez', 'Castro', 'Pedro', 'pedro.gomez@email.com',
'Callejón del Sol 202', 'Piura', 'Piura', '987654325', '1987-12-18', '2024-05-22
12:30:00'),
('CE', '20123456789012345681', 'Sánchez', 'Rodríguez', 'Marta',
'marta.sanchez@email.com', 'Calle 123', 'Lima', 'Lima', '987654326', '1989-01-05',
'2024-06-07 09:00:00'),
('DNI', '34567891', 'Martínez', 'Jiménez', 'Carlos', 'carlos.martinez@email.com', 'Jr.
de la Paz 204', 'Chiclayo', 'Lambayeque', '987654327', '1995-07-22', '2024-07-14
19:45:00'),
('DNI', '56789013', 'Vásquez', 'Mendoza', 'Roberto', 'roberto.vasquez@email.com', 'Av.
Libertad 303', 'Ica', 'Ica', '987654328', '1990-11-30', '2024-08-05 10:30:00'),
('CE', '20123456789012345682', 'Torres', 'Salazar', 'Elena', 'elena.torres@email.com',
'Calle Angosta 405', 'Tacna', 'Tacna', '987654329', '1991-09-15', '2024-09-20
13:00:00'),
('DNI', '65432110', 'Mendoza', 'Guerra', 'José', 'jose.mendoza@email.com', 'Calle Los
Álamos 506', 'Lima', 'Lima', '987654330', '1994-11-10', '2024-10-15 14:00:00'),
('DNI', '23456791', 'Silva', 'Jara', 'Lucía', 'lucia.silva@email.com', 'Av. Las Rosas
607', 'Arequipa', 'Arequipa', '987654331', '1992-06-03', '2024-11-02 15:30:00'),
('CE', '20123456789012345683', 'Cáceres', 'Ramírez', 'Marta',
'marta.caceres@email.com', 'Jr. La Paz 708', 'Iquitos', 'Loreto', '987654332', '1993-
01-12', '2024-12-04 16:00:00'),
('DNI', '12345679', 'García', 'Valencia', 'Ricardo', 'ricardo.garcia@email.com', 'Av.
del Mar 809', 'Piura', 'Piura', '987654333', '1990-03-15', '2024-01-30 09:45:00'),
('CE', '20123456789012345684', 'Aguirre', 'Hernández', 'Juan',
'juan.aguirre@email.com', 'Calle Fama 910', 'Chimbote', 'Áncash', '987654334', '1991-
05-20', '2024-02-15 17:30:00'),
('CE', '20123456789012345685', 'Romero', 'López', 'Andrea', 'andrea.romero@email.com',
'Jr. Central 1111', 'Cusco', 'Cusco', '987654335', '1994-10-11', '2024-03-25
18:00:00'),
('DNI', '45678902', 'Quispe', 'Maldonado', 'Carlos', 'carlos.quispe@email.com', 'Calle
Nueva 1122', 'Lima', 'Lima', '987654336', '1992-07-18', '2024-04-20 08:30:00');
```

Registros para la tabla CATEGORÍAS:

Para efectos didácticos, se especificó en el código cuáles son categorías y cuáles son subcategorías.

```
INSERT INTO categorias (nombre, categoria_padre_id)
VALUES
('Electrónica', NULL),    -- Categoría padre
('Celulares y Accesorios', 1), -- Subcategoría de Electrónica
('Laptops y Computadoras', 1), -- Subcategoría de Electrónica
('Televisores y Audio', 1), -- Subcategoría de Electrónica
('Ropa y Moda', NULL),    -- Categoría padre
('Hombres', 5),           -- Subcategoría de Ropa y Moda
('Mujeres', 5),           -- Subcategoría de Ropa y Moda
('Calzado', 5),           -- Subcategoría de Ropa y Moda
('Hogar y Muebles', NULL), -- Categoría padre
('Muebles para Sala', 9), -- Subcategoría de Hogar y Muebles
('Muebles para Cocina', 9), -- Subcategoría de Hogar y Muebles
('Deportes y Fitness', NULL), -- Categoría padre

('Equipos de Gimnasio', 12); -- Subcategoría de Deportes y Fitness
```

Registros para la tabla PRODUCTOS:

Similar al caso anterior, para descripción se ingresaron los productos por categoría y subcategoría a las que pertenecen, especificando ello.

```
-- Electrónica -> Celulares y Accesorios
('iPhone 15', 'Smartphone de última generación con cámara de 48 MP', 899.99, 100, 2),
('Samsung Galaxy S23', 'Smartphone con pantalla AMOLED y cámara de 50 MP', 800.00,
120, 2),
('AirPods Pro 2', 'Auriculares inalámbricos con cancelación activa de ruido', 249.99,
150, 2),
('Samsung Galaxy Buds 2', 'Auriculares inalámbricos con sonido dinámico', 149.99, 200,
2),
('Funda OtterBox para iPhone 15', 'Funda protectora de alta calidad para iPhone',
45.99, 250, 2),
('Samsung Galaxy Watch 5', 'Reloj inteligente con monitorización de salud', 229.99,
90, 2),

-- Electrónica -> Laptops y Computadoras
('MacBook Pro 14"', 'Laptop de alto rendimiento con chip M1 Pro', 1999.99, 60, 3),
('Dell XPS 13', 'Laptop ultradelgada con pantalla 4K y procesador i7', 1499.00, 70,
3),
('HP Pavilion 15', 'Laptop con procesador Intel Core i5 y pantalla Full HD', 649.99,
150, 3),
('Lenovo ThinkPad X1', 'Laptop empresarial con pantalla 4K y alto rendimiento',
1799.00, 40, 3),
('Microsoft Surface Laptop 4', 'Laptop premium con pantalla táctil y procesador Intel
i7', 1299.99, 80, 3),
('Asus VivoBook 15', 'Laptop asequible para estudiantes y profesionales', 499.99, 180,
3),
('Acer Predator Helios 300', 'Laptop para juegos con pantalla 144 Hz y RTX 3060',
1699.00, 120, 3),

-- Electrónica -> Televisores y Audio
('Samsung 65" 4K UHD TV', 'Televisor Samsung UHD con Smart TV integrado', 899.00, 50,
4),
('LG OLED 55"', 'Televisor OLED con calidad de imagen inigualable', 1299.00, 40, 4),
```

```

('Sonos Beam', 'Barra de sonido para mejorar la experiencia de audio en el hogar',
399.99, 200, 4),
('Bose SoundLink Revolve+', 'Altavoz Bluetooth portátil con sonido envolvente',
299.00, 300, 4),

-- Ropa y Moda -> Hombres
('Camiseta Nike Dri-Fit', 'Camiseta deportiva de secado rápido', 29.99, 150, 5),
('Pantalones Levi's 501', 'Pantalones de mezclilla clásicos para hombre', 79.99, 100,
5),
('Chaqueta Columbia', 'Chaqueta impermeable para clima frío', 99.00, 80, 5),
('Zapatos Adidas Ultraboost', 'Zapatos deportivos cómodos y de alto rendimiento',
119.00, 200, 5),
('Reloj Casio G-Shock', 'Reloj resistente con múltiples funciones para hombres',
79.00, 300, 5),

-- Ropa y Moda -> Mujeres
('Blusa H&M', 'Blusa casual de manga corta para mujeres', 24.99, 150, 6),
('Vestido Zara', 'Vestido elegante para ocasiones especiales', 49.99, 120, 6),
('Zapatos Boda Jimmy Choo', 'Zapatos elegantes para bodas y eventos', 449.00, 80, 6),
('Pantalones Stradivarius', 'Pantalones cómodos para el día a día', 39.99, 200, 6),
('Cartera Michael Kors', 'Cartera de cuero de alta calidad', 149.99, 50, 6),

-- Ropa y Moda -> Calzado
('Nike Air Max', 'Zapatillas deportivas con diseño innovador', 89.99, 300, 7),
('Adidas Stan Smith', 'Zapatillas clásicas de tenis para hombres y mujeres', 79.00,
250, 7),
('Botines Timberland', 'Botines de cuero ideales para clima frío', 119.99, 100, 7),
('Sandalias Puma', 'Sandalias cómodas para el verano', 34.99, 150, 7),

-- Hogar y Muebles -> Muebles para Sala
('Sofá Sealy', 'Sofá de tres puestos con cojines cómodos', 499.00, 30, 8),
('Mesa de Centro Ikea', 'Mesa de centro moderna y funcional', 129.00, 60, 8),
('Sillón reclinable Homeline', 'Sillón de piel con reclinación automática', 299.99,
40, 8),
('Estantería modular Amazon Basics', 'Estantería de madera para libros y objetos',
89.99, 100, 8),

-- Hogar y Muebles -> Muebles para Cocina
('Mesa comedor madera', 'Mesa de comedor de 6 puestos, madera de pino', 199.00, 70,
9),
('Alacena Ikea', 'Alacena para almacenar alimentos y utensilios', 79.00, 80, 9),
('Cocinero de gas Whirlpool', 'Cocinero a gas con 4 hornillas y horno', 399.99, 50,
9),

-- Deportes y Fitness -> Equipos de Gimnasio
('Bicicleta Estática', 'Bicicleta estática con 12 niveles de resistencia', 199.00,
120, 12),
('Mancuernas Ajustables', 'Juego de mancuernas de peso ajustable', 79.00, 200, 12),

```

Registros para la tabla TIENDAS:

```

INSERT INTO tiendas (nombre, direccion, ciudad, departamento)
VALUES('Tienda Lima Centro', 'Av. Pardo y Aliaga 100', 'Lima', 'Lima'),
('Tienda San Isidro', 'Av. Juan de Arona 500', 'San Isidro', 'Lima'),
('Tienda Miraflores', 'Av. Pardo 900', 'Miraflores', 'Lima'),
('Tienda Barranco', 'Av. Pedro de Osma 650', 'Barranco', 'Lima'),
('Tienda Callao', 'Av. Oscar R. Benavides 1500', 'Callao', 'Callao'),
('Tienda Arequipa', 'Calle San José 200', 'Arequipa', 'Arequipa'),
('Tienda Trujillo', 'Av. Nicolás de Piérola 300', 'Trujillo', 'La Libertad'),
('Tienda Cusco', 'Calle Plateros 400', 'Cusco', 'Cusco'),

```

```
('Tienda Piura', 'Av. Sánchez Cerro 500', 'Piura', 'Piura'),  
('Tienda Chiclayo', 'Calle Independencia 600', 'Chiclayo', 'Lambayeque');
```

Registros para la tabla PEDIDOS:

```
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (10, 5, '2024-12-05 11:30:00', 279.80, 'Pendiente'); -- Producto ID: 1, 2  
unidades de producto (139.90 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (14, 1, '2024-12-06 12:00:00', 459.90, 'Enviado'); -- Producto ID: 3, 3  
unidades de producto (153.30 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (7, 8, '2024-12-07 14:00:00', 419.90, 'Entregado'); -- Producto ID: 5, 4  
unidades de producto (104.98 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (1, 2, '2024-12-08 09:45:00', 279.90, 'Enviado'); -- Producto ID: 2, 3 unidades  
de producto (93.30 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (19, 4, '2024-12-08 10:30:00', 249.80, 'Pendiente'); -- Producto ID: 4, 2  
unidades de producto (124.90 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (13, 7, '2024-12-09 13:15:00', 409.80, 'Enviado'); -- Producto ID: 6, 4  
unidades de producto (102.45 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (3, 9, '2024-12-09 15:00:00', 179.90, 'Pendiente'); -- Producto ID: 8, 1 unidad  
de producto (179.90 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (5, 10, '2024-12-10 09:30:00', 359.80, 'Enviado'); -- Producto ID: 10, 4  
unidades de producto (89.95 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (18, 3, '2024-12-11 11:20:00', 349.90, 'Pendiente'); -- Producto ID: 9, 3  
unidades de producto (116.63 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (9, 6, '2024-12-12 12:00:00', 149.90, 'Enviado'); -- Producto ID: 11, 1 unidad  
de producto (149.90 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (2, 4, '2024-12-12 13:45:00', 449.70, 'Pendiente'); -- Producto ID: 4, 3  
unidades de producto (149.90 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (20, 5, '2024-12-13 10:00:00', 209.90, 'Enviado'); -- Producto ID: 12, 2  
unidades de producto (104.95 c/u)  
INSERT INTO pedidos (cliente_id, tienda_id, fecha_pedido, total, estado)  
VALUES (17, 8, '2024-12-13 12:00:00', 329.90, 'Entregado'); -- Producto ID: 13, 4  
unidades de producto (82.47 c/u);
```

Registros para la tabla DETALLES DE PEDIDOS:

```
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)  
VALUES (1, 1, 2, 279.80);  
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)  
VALUES (2, 3, 3, 459.90);  
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)  
VALUES (3, 5, 4, 419.92);  
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)  
VALUES (4, 2, 3, 279.90);  
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)  
VALUES (5, 4, 2, 249.80);
```

```

INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (6, 6, 4, 409.80);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (7, 8, 1, 179.90);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (8, 10, 4, 359.80);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (9, 9, 3, 349.89);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (10, 11, 1, 149.90);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (11, 4, 3, 374.70);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (12, 12, 2, 209.90);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (13, 13, 4, 329.88);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (14, 3, 2, 259.80);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (15, 7, 1, 189.90);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (16, 5, 3, 359.91);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (17, 4, 3, 299.91);
INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad, precio)
VALUES (18, 6, 1, 169.90);

```

Registros para la tabla MÉTODOS DE PAGO:

```

INSERT INTO metodo_pago (nombre)
VALUES
('Tarjeta de crédito'),
('Tarjeta de débito'),
('QR Yape/Plin'),
('Banca por internet'),
('PayPal'),
('Bitcoin'),
('Transferencia bancaria');

```

Registros para la tabla PAGOS:

```

INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (1, 4, 279.80, 'Confirmado', '2024-12-05 11:30:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (2, 1, 459.90, 'Confirmado', '2024-12-06 12:00:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (3, 2, 419.90, 'En Proceso', '2024-12-07 14:00:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (4, 3, 279.90, 'Confirmado', '2024-12-08 09:45:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (5, 7, 249.80, 'Confirmado', '2024-12-08 10:30:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (6, 4, 409.80, 'En Proceso', '2024-12-09 13:15:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (7, 5, 179.90, 'Rechazado', '2024-12-09 15:00:00');
INSERT INTO Pagos (pedido_id, metodo_pago_id, importe, estado, fecha_pago)
VALUES (8, 1, 359.80, 'Confirmado', '2024-12-10 09:30:00');

```

Registros para la tabla ENVÍOS:

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (1, 'Av. Pardo y Aliaga 123', 'Lima', 'Lima', 'Referencia 1', 'En Preparación',
'2024-12-05 11:30:00', NULL);
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (2, 'Calle Las Flores 456', 'Arequipa', 'Arequipa', 'Referencia 2', 'En
Preparación', '2024-12-06 12:00:00', NULL);
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (3, 'Jr. Los Olivos 789', 'Cusco', 'Cusco', 'Referencia 3', 'En Preparación',
'2024-12-07 14:00:00', NULL);
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (4, 'Av. Universitaria 101', 'Trujillo', 'La Libertad', 'Referencia 4', 'En
Tránsito', '2024-12-08 09:45:00', '2024-12-11 09:45:00');
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (5, 'Calle Las Palmas 222', 'Lima', 'Lima', 'Referencia 5', 'En Preparación',
'2024-12-08 10:30:00', NULL);
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (6, 'Jr. San Martín 444', 'Arequipa', 'Arequipa', 'Referencia 6', 'En
Preparación', '2024-12-09 13:15:00', NULL);
```

```
INSERT INTO envios (pedido_id, direccion_envio, ciudad_envio, departamento,
referencia, estado_envio, fecha_envio, fecha_entrega)
VALUES (7, 'Av. Javier Prado 567', 'Lima', 'Lima', 'Referencia 7', 'Rechazado', '2024-
12-09 15:00:00', NULL);
```

Registros para la tabla STOCK DE TIENDA:

Para este caso, se mostrará el ingreso de información para las tres primeras tiendas.

```
INSERT INTO stock_tienda (producto_id, tienda_id, cantidad) VALUES
(1, 1, 15), (2, 1, 20), (3, 1, 25),
(1, 2, 10), (2, 2, 15), (3, 2, 20),
(1, 3, 20), (2, 3, 25), (3, 3, 30),
(1, 4, 10), (2, 4, 20), (3, 4, 40),
(1, 5, 5), (2, 5, 10), (3, 5, 15),
(1, 6, 5), (2, 6, 10), (3, 6, 10),
(1, 7, 10), (2, 7, 5), (3, 7, 10),
(1, 8, 5), (2, 8, 10), (3, 8, 10),
(1, 9, 10), (2, 9, 15), (3, 9, 10);
(1, 10, 10), (2, 10, 10),
```


VI. CONSULTAS DESARROLLADAS PARA EL ANÁLISIS DE DATOS

Las siguientes consultas SQL se han desarrollado para extraer información relevante del sistema de base de datos. Estas consultas son útiles para generar reportes y análisis de alto valor para las partes interesadas del negocio. Tomar en cuenta que la información ingresada corresponde solo al mes de diciembre 2024, por lo que se estaría realizando un análisis de dicho periodo para evaluar el avance del negocio.

1. Total de pedidos realizados

Esta consulta muestra el número total de pedidos realizados en el sistema. Es útil para obtener una visión general de la actividad en la tienda.

```
SELECT COUNT(*) AS Total_Pedidos
FROM pedidos;
```

| Results | | Messages |
|---------------|----|----------|
| Total_Pedidos | | |
| 1 | 42 | |

2. Productos con stock bajo

Esta consulta devuelve los productos cuyo stock es menor que un umbral definido. Es útil para monitorear la disponibilidad de productos. Como parámetro, se considera stock bajo un total de 50 unidades o menos a nivel de todas las tiendas.

```
SELECT nombre, stock
FROM productos
WHERE stock < 50;
```

| | nombre | stock |
|---|-----------------------------|-------|
| 1 | Lenovo ThinkPad X1 | 40 |
| 2 | LG OLED 55" | 40 |
| 3 | Sofá Sealy | 30 |
| 4 | Sillón reclinable Homeline | 40 |
| 5 | Cinta de Correr NordicTrack | 30 |

3. Información de clientes recientes

Esta consulta muestra una lista de los clientes registrados en los últimos 10 días, lo cual es útil para monitorear la adquisición de nuevos clientes.

```
SELECT id, nombre, apellido_paterno, apellido_materno, email, fecha_registro
FROM clientes
WHERE fecha_registro >= DATEADD(DAY, -10, GETDATE());
```

| | id | nombre | apellido_paterno | apellido_materno | email | fecha_registro |
|---|----|--------|------------------|------------------|---------------------------|-------------------------|
| 1 | 24 | Lucía | Maldonado | Alvarado | lucia.maldonado@email.com | 2024-12-20 14:30:00.000 |

4. Total de ventas por tienda

Esta consulta muestra el total de ventas generadas por cada tienda en el sistema, ayudando a analizar el rendimiento de cada una. Tomar en cuenta que no se especifica el periodo, pues solo se cuenta con información de diciembre 2024.

```
SELECT t.nombre AS Tienda, SUM(p.total) AS Total_Ventas
FROM pedidos p
JOIN tiendas t ON p.tienda_id = t.id
GROUP BY t.nombre
ORDER BY total_ventas DESC
```

| | Tienda | Total_Ventas |
|----|--------------------|--------------|
| 1 | Tienda Miraflores | 1519.30 |
| 2 | Tienda Barranco | 1499.00 |
| 3 | Tienda Cusco | 1469.50 |
| 4 | Tienda Trujillo | 1359.40 |
| 5 | Tienda Arequipa | 1349.40 |
| 6 | Tienda San Isidro | 1279.50 |
| 7 | Tienda Piura | 1209.30 |
| 8 | Tienda Callao | 889.60 |
| 9 | Tienda Chiclayo | 879.60 |
| 10 | Tienda Lima Centro | 639.80 |

5. Productos más vendidos (por cantidad)

Esta consulta permite obtener los productos más vendidos, ordenados por la cantidad vendida. Es útil para saber qué productos están siendo más demandados.

```
SELECT pr.nombre AS Producto, SUM(dp.cantidad) AS Cantidad_Vendida
FROM detalles_pedido dp
JOIN productos pr ON dp.producto_id = pr.id
GROUP BY pr.nombre
ORDER BY Cantidad_Vendida DESC;
```

| | Producto | Cantidad_Vendida |
|----|-------------------------------|------------------|
| 1 | Samsung Galaxy Buds 2 | 12 |
| 2 | Xiaomi Redmi Note 12 | 12 |
| 3 | Cargador inalámbrico Anker | 10 |
| 4 | MacBook Pro 14" | 10 |
| 5 | iPhone 15 | 9 |
| 6 | AirPods Pro 2 | 9 |
| 7 | Microsoft Surface Laptop 4 | 8 |
| 8 | Lenovo ThinkPad X1 | 7 |
| 9 | Dell XPS 13 | 6 |
| 10 | Samsung Galaxy S23 | 5 |
| 11 | Samsung Galaxy Watch 5 | 4 |
| 12 | Funda OtterBox para iPhone 15 | 2 |
| 13 | HP Pavilion 15 | 2 |

6. Detalles de pagos confirmados

Esta consulta muestra todos los pagos confirmados, incluyendo el pedido relacionado, el método de pago, el importe y la fecha de pago. Es útil para auditar los pagos procesados.

```
SELECT p.id AS Pago_ID, p.importe, mp.nombre AS Metodo_Pago, p.fecha_pago,
pe.id AS Pedido_ID
FROM pagos p
JOIN metodo_pago mp ON p.metodo_pago_id = mp.id
JOIN pedidos pe ON p.pedido_id = pe.id
WHERE p.estado = 'Confirmado';
```

| | Pago_ID | importe | Metodo_Pago | fecha_pago | Pedido_ID |
|----|---------|---------|------------------------|-------------------------|-----------|
| 1 | 1 | 279.80 | Banca por internet | 2024-12-05 11:30:00.000 | 1 |
| 2 | 2 | 459.90 | Tarjeta de crédito | 2024-12-06 12:00:00.000 | 2 |
| 3 | 4 | 279.90 | QR Yape/Plin | 2024-12-08 09:45:00.000 | 4 |
| 4 | 5 | 249.80 | Transferencia bancaria | 2024-12-08 10:30:00.000 | 5 |
| 5 | 8 | 359.80 | Tarjeta de crédito | 2024-12-10 09:30:00.000 | 8 |
| 6 | 9 | 349.90 | Tarjeta de débito | 2024-12-11 11:20:00.000 | 9 |
| 7 | 11 | 449.70 | Bitcoin | 2024-12-12 13:45:00.000 | 11 |
| 8 | 12 | 209.90 | PayPal | 2024-12-13 10:00:00.000 | 12 |
| 9 | 14 | 259.80 | QR Yape/Plin | 2024-12-14 09:30:00.000 | 14 |
| 10 | 16 | 359.90 | Tarjeta de débito | 2024-12-14 13:00:00.000 | 16 |
| 11 | 17 | 299.90 | PayPal | 2024-12-15 10:40:00.000 | 17 |
| 12 | 19 | 219.90 | Transferencia bancaria | 2024-12-16 11:00:00.000 | 19 |
| 13 | 20 | 249.80 | QR Yape/Plin | 2024-12-17 12:15:00.000 | 20 |
| 14 | 21 | 269.80 | Tarjeta de crédito | 2024-12-18 10:10:00.000 | 21 |
| 15 | 22 | 399.90 | Tarjeta de débito | 2024-12-18 13:30:00.000 | 22 |
| 16 | 24 | 329.90 | PayPal | 2024-12-19 12:15:00.000 | 24 |
| 17 | 26 | 179.90 | Tarjeta de crédito | 2024-12-20 09:30:00.000 | 26 |
| 18 | 28 | 269.80 | Tarjeta de débito | 2024-12-21 13:30:00.000 | 28 |
| 19 | 29 | 249.90 | Banca por internet | 2024-12-22 11:00:00.000 | 29 |
| 20 | 30 | 359.80 | Transferencia bancaria | 2024-12-23 10:20:00.000 | 30 |
| 21 | 32 | 209.90 | Tarjeta de crédito | 2024-12-24 12:30:00.000 | 32 |
| 22 | 33 | 179.90 | Bitcoin | 2024-12-25 14:30:00.000 | 33 |
| 23 | 36 | 219.90 | PayPal | 2024-12-27 10:30:00.000 | 36 |
| 24 | 37 | 449.90 | Tarjeta de crédito | 2024-12-28 09:10:00.000 | 37 |
| 25 | 39 | 269.80 | QR Yape/Plin | 2024-12-30 11:00:00.000 | 39 |
| 26 | 42 | 269.80 | Tarjeta de débito | 2024-12-31 16:00:00.000 | 42 |

7. Ventas por fecha específica

Esta consulta muestra las ventas totales agrupadas por mes y año, lo cual es útil para identificar tendencias de ventas y patrones estacionales. Por ahora solo se tiene diciembre 2024, pero serviría para analizar un periodo de tiempo mayor.

```
SELECT YEAR(fecha_pedido) AS Año, MONTH(fecha_pedido) AS Mes, SUM(total) AS
Total_Ventas
FROM pedidos
GROUP BY YEAR(fecha_pedido), MONTH(fecha_pedido)
ORDER BY Año DESC, Mes DESC;
```

| | Año | Mes | Total_Ventas |
|---|------|-----|--------------|
| 1 | 2024 | 12 | 12094.40 |

8. Análisis de stock por tienda y producto

Esta consulta muestra el stock disponible de cada producto por tienda. Es útil para gestionar el inventario y monitorear el stock en cada ubicación.

```
SELECT t.nombre AS Tienda, pr.nombre AS Producto, st.cantidad AS
Stock_Disponible
FROM stock_tienda st
JOIN tiendas t ON st.tienda_id = t.id
JOIN productos pr ON st.producto_id = pr.id
ORDER BY t.nombre, pr.nombre;
```

| | Tienda | Producto | Stock_Disponible |
|----|-----------------|----------------------------------|------------------|
| 1 | Tienda Arequipa | Acer Predator Helios 300 | 5 |
| 2 | Tienda Arequipa | Adidas Stan Smith | 20 |
| 3 | Tienda Arequipa | AirPods Pro 2 | 10 |
| 4 | Tienda Arequipa | Alacena Ikea | 10 |
| 5 | Tienda Arequipa | Asus VivoBook 15 | 20 |
| 6 | Tienda Arequipa | Bicicleta Estática | 10 |
| 7 | Tienda Arequipa | Blusa H&M | 20 |
| 8 | Tienda Arequipa | Bose SoundLink Revolve+ | 20 |
| 9 | Tienda Arequipa | Botines Timberland | 5 |
| 10 | Tienda Arequipa | Camiseta Nike Dri-Fit | 20 |
| 11 | Tienda Arequipa | Cargador inalámbrico Anker | 30 |
| 12 | Tienda Arequipa | Cartera Michael Kors | 10 |
| 13 | Tienda Arequipa | Chaqueta Columbia | 10 |
| 14 | Tienda Arequipa | Cinta de Correr NordicTrack | 5 |
| 15 | Tienda Arequipa | Cocinero de gas Whirlpool | 5 |
| 16 | Tienda Arequipa | Dell XPS 13 | 10 |
| 17 | Tienda Arequipa | Estantería modular Amazon Basics | 10 |
| 18 | Tienda Arequipa | Funda OtterBox para iPhone 15 | 30 |
| 19 | Tienda Arequipa | HP Pavilion 15 | 10 |
| 20 | Tienda Arequipa | iPhone 15 | 5 |

9. Análisis de envíos pendientes y entregados por clientes

Esta consulta muestra la información sobre los envíos pendientes y entregados, agrupados por cliente. Es útil para el seguimiento de los pedidos y el estado de los envíos.

```
SELECT c.nombre AS Cliente, c.apellido_paterno, e.estado_envio, COUNT(e.id) AS
Cantidad_Envios
FROM envios e
JOIN pedidos p ON e.pedido_id = p.id
JOIN clientes c ON p.cliente_id = c.id
GROUP BY c.nombre, c.apellido_paterno, e.estado_envio
ORDER BY c.nombre, e.estado_envio;
```

| | Ciente | apellido_paterno | estado_envio | Cantidad_Envios |
|----|---------|------------------|----------------|-----------------|
| 5 | Eduardo | Salinas | En Preparación | 1 |
| 6 | Elena | Torres | En Preparación | 1 |
| 7 | José | Mendoza | En Preparación | 1 |
| 8 | José | Ríos | En Preparación | 1 |
| 9 | Juan | Aguirre | En Preparación | 1 |
| 10 | Juan | Pérez | En Tránsito | 1 |
| 11 | Lucía | Maldonado | En Tránsito | 1 |
| 12 | Lucía | Silva | Rechazado | 1 |
| 13 | María | Campos | En Preparación | 1 |
| 14 | Miguel | Vargas | En Tránsito | 1 |
| 15 | Natalia | Vega | En Preparación | 1 |
| 16 | Pedro | Gómez | En Preparación | 1 |
| 17 | Pedro | Gómez | En Tránsito | 1 |
| 18 | Ricardo | García | En Preparación | 1 |
| 19 | Ricardo | Ramírez | En Preparación | 1 |
| 20 | Roberto | Vásquez | En Preparación | 1 |

10. Análisis de rentabilidad por producto

Esta consulta muestra los productos más rentables para la tienda, basándose en el precio de venta y la cantidad vendida. Ayuda a identificar qué productos generan mayores ingresos para el negocio.

```
SELECT pr.nombre AS Producto,
       SUM(dp.cantidad) AS Cantidad_Vendida,
       SUM(dp.cantidad * dp.precio) AS Ingresos_Generados,
       (SUM(dp.cantidad * dp.precio) / SUM(dp.cantidad)) AS Precio_Promedio
FROM detalles_pedido dp
JOIN productos pr ON dp.producto_id = pr.id
GROUP BY pr.nombre
ORDER BY Ingresos_Generados DESC;
```

| | Producto | Cantidad_Vendida | Ingresos_Generados | Precio_Promedio |
|----|-------------------------------|------------------|--------------------|-----------------|
| 1 | Xiaomi Redmi Note 12 | 12 | 4498.94 | 374.911666 |
| 2 | Cargador inalámbrico Anker | 10 | 3608.10 | 360.810000 |
| 3 | Samsung Galaxy Buds 2 | 12 | 3542.83 | 295.235833 |
| 4 | MacBook Pro 14" | 10 | 3388.97 | 338.897000 |
| 5 | AirPods Pro 2 | 9 | 3018.93 | 335.436666 |
| 6 | Microsoft Surface Laptop 4 | 8 | 2639.20 | 329.900000 |
| 7 | iPhone 15 | 9 | 2398.93 | 266.547777 |
| 8 | Dell XPS 13 | 6 | 1979.00 | 329.833333 |
| 9 | Lenovo ThinkPad X1 | 7 | 1678.90 | 239.842857 |
| 10 | Samsung Galaxy S23 | 5 | 1379.30 | 275.860000 |
| 11 | Samsung Galaxy Watch 5 | 4 | 899.40 | 224.850000 |
| 12 | Funda OtterBox para iPhone 15 | 2 | 369.80 | 184.900000 |
| 13 | HP Pavilion 15 | 2 | 329.80 | 164.900000 |

11. Análisis de clientes frecuentes

Esta consulta muestra los clientes que han realizado más compras en el sistema, con el número total de pedidos y el total gastado por cada cliente. Es útil para identificar a los clientes más valiosos para la tienda, especificando que se necesita el dato de clientes que hayan realizado más de un pedido en el periodo analizado.

```

SELECT c.id AS Cliente_ID,
       c.nombre AS Cliente,
       c.apellido_paterno,
       COUNT(p.id) AS Total_Pedidos,
       SUM(p.total) AS Total_Gastado
FROM clientes c
JOIN pedidos p ON c.id = p.cliente_id
GROUP BY c.id, c.nombre, c.apellido_paterno
HAVING COUNT(p.id) > 1 -- Solo mostrar clientes con más de un pedido
ORDER BY Total_Gastado DESC;

```

| | Cliente_ID | Cliente | apellido_paterno | Total_Pedidos | Total_Gastado |
|----|------------|---------|------------------|---------------|---------------|
| 1 | 2 | Carlos | Ramírez | 3 | 1029.40 |
| 2 | 5 | Pedro | Gómez | 3 | 909.60 |
| 3 | 13 | Ricardo | García | 2 | 829.60 |
| 4 | 7 | Carlos | Martínez | 2 | 779.80 |
| 5 | 16 | Carlos | Quispe | 2 | 699.70 |
| 6 | 12 | Marta | Cáceres | 2 | 669.70 |
| 7 | 4 | Luis | Fernández | 2 | 599.80 |
| 8 | 1 | Juan | Pérez | 2 | 579.80 |
| 9 | 23 | José | Cordero | 2 | 569.80 |
| 10 | 18 | Miguel | Vargas | 2 | 529.80 |
| 11 | 15 | Andrea | Romero | 2 | 479.70 |
| 12 | 11 | Lucía | Silva | 2 | 459.70 |
| 13 | 3 | Ana | Lopez | 2 | 449.70 |
| 14 | 19 | Eduardo | Salinas | 2 | 429.70 |
| 15 | 6 | Marta | Sánchez | 2 | 399.80 |
| 16 | 9 | Elena | Torres | 2 | 389.80 |

12. Análisis de ventas mensuales por tienda (aplicaría para periodos más amplios)

Esta consulta muestra las ventas totales por tienda, agrupadas por mes y año, lo que permite hacer un análisis temporal del rendimiento de las tiendas.

```

SELECT t.nombre AS Tienda,
       YEAR(p.fecha_pedido) AS Año,
       MONTH(p.fecha_pedido) AS Mes,
       SUM(p.total) AS Total_Ventas
FROM pedidos p
JOIN tiendas t ON p.tienda_id = t.id
GROUP BY t.nombre, YEAR(p.fecha_pedido), MONTH(p.fecha_pedido)
ORDER BY Tienda, Año DESC, Mes DESC;

```

| | Tienda | Año | Mes | Total_Ventas |
|----|--------------------|------|-----|--------------|
| 1 | Tienda Arequipa | 2024 | 12 | 1349.40 |
| 2 | Tienda Barranco | 2024 | 12 | 1499.00 |
| 3 | Tienda Callao | 2024 | 12 | 889.60 |
| 4 | Tienda Chiclayo | 2024 | 12 | 879.60 |
| 5 | Tienda Cusco | 2024 | 12 | 1469.50 |
| 6 | Tienda Lima Centro | 2024 | 12 | 639.80 |
| 7 | Tienda Miraflores | 2024 | 12 | 1519.30 |
| 8 | Tienda Piura | 2024 | 12 | 1209.30 |
| 9 | Tienda San Isidro | 2024 | 12 | 1279.50 |
| 10 | Tienda Trujillo | 2024 | 12 | 1359.40 |

13. Comparación de productos más vendidos por categoría

Esta consulta muestra una comparación de los productos más vendidos dentro de cada categoría. Permite identificar cuáles son los productos más populares por categoría y ver las tendencias de ventas dentro de las distintas secciones del catálogo.

```
SELECT c.nombre AS Categoria,
       pr.nombre AS Producto,
       SUM(dp.cantidad) AS Cantidad_Vendida
FROM detalles_pedido dp
JOIN productos pr ON dp.producto_id = pr.id
JOIN categorias c ON pr.categoria_id = c.id
GROUP BY c.nombre, pr.nombre
ORDER BY c.nombre, Cantidad_Vendida DESC;
```

| | Categoria | Producto | Cantidad_Vendida |
|----|------------------------|-------------------------------|------------------|
| 1 | Celulares y Accesorios | Samsung Galaxy Buds 2 | 12 |
| 2 | Celulares y Accesorios | Xiaomi Redmi Note 12 | 12 |
| 3 | Celulares y Accesorios | Cargador inalámbrico Anker | 10 |
| 4 | Celulares y Accesorios | iPhone 15 | 9 |
| 5 | Celulares y Accesorios | AirPods Pro 2 | 9 |
| 6 | Celulares y Accesorios | Samsung Galaxy S23 | 5 |
| 7 | Celulares y Accesorios | Samsung Galaxy Watch 5 | 4 |
| 8 | Celulares y Accesorios | Funda OtterBox para iPhone 15 | 2 |
| 9 | Laptops y Computadoras | MacBook Pro 14" | 10 |
| 10 | Laptops y Computadoras | Microsoft Surface Laptop 4 | 8 |
| 11 | Laptops y Computadoras | Lenovo ThinkPad X1 | 7 |
| 12 | Laptops y Computadoras | Dell XPS 13 | 6 |
| 13 | Laptops y Computadoras | HP Pavilion 15 | 2 |

14. Análisis de productos sin stock por tienda

Esta consulta muestra los productos que no tienen stock en cada tienda/almacén. Ayuda a identificar rápidamente los productos agotados y tomar acciones para reabastecer el inventario. En este caso, todas las tiendas cuentan con stock.

```
SELECT t.nombre AS Tienda,
       pr.nombre AS Producto,
       st.cantidad AS Stock_Disponible
FROM stock_tienda st
JOIN tiendas t ON st.tienda_id = t.id
JOIN productos pr ON st.producto_id = pr.id
WHERE st.cantidad = 0
ORDER BY t.nombre, pr.nombre;
```

| Tienda | Producto | Stock_Disponible |
|--------|----------|------------------|
|--------|----------|------------------|

15. Promedio de tiempo de envío por tienda

Esta consulta calcula el promedio de tiempo que tarda en realizarse el envío de un pedido, por tienda. El resultado es útil para evaluar la eficiencia de cada tienda en el proceso de envío. Se consideran solo aquellas tiendas que ya entregaron pedidos.

```
SELECT t.nombre AS Tienda,  
       AVG(DATEDIFF(DAY, e.fecha_envio, e.fecha_entrega)) AS  
Promedio_Tiempo_Envio  
FROM envios e  
JOIN tiendas t ON e.pedido_id IN (SELECT id FROM pedidos WHERE tienda_id =  
t.id)  
WHERE e.fecha_entrega IS NOT NULL  
GROUP BY t.nombre  
ORDER BY Promedio_Tiempo_Envio DESC;
```

| | Tienda | Promedio_Tiempo_Envio |
|---|-------------------|-----------------------|
| 1 | Tienda Barranco | 3 |
| 2 | Tienda Miraflores | 3 |
| 3 | Tienda San Isidro | 3 |

16. Análisis de pedidos por estado de envío

Esta consulta muestra la cantidad de pedidos que se encuentran en cada estado de envío (por ejemplo, "En Preparación", "En Tránsito", "Entregado") en cada tienda. Es útil para conocer el estado de los pedidos y la eficiencia en las entregas.

```
SELECT t.nombre AS Tienda,  
       e.estado_envio AS Estado_Envio,  
       COUNT(e.id) AS Cantidad_Pedidos  
FROM envios e  
JOIN pedidos p ON e.pedido_id = p.id  
JOIN tiendas t ON p.tienda_id = t.id  
GROUP BY t.nombre, e.estado_envio  
ORDER BY t.nombre, Estado_Envio;
```

| | Tienda | Estado_Envio | Cantidad_Pedidos |
|----|--------------------|----------------|------------------|
| 1 | Tienda Arequipa | En Preparación | 2 |
| 2 | Tienda Barranco | En Preparación | 2 |
| 3 | Tienda Barranco | En Tránsito | 1 |
| 4 | Tienda Callao | En Preparación | 2 |
| 5 | Tienda Chiclayo | En Preparación | 2 |
| 6 | Tienda Cusco | En Preparación | 2 |
| 7 | Tienda Lima Centro | En Preparación | 1 |
| 8 | Tienda Miraflores | En Tránsito | 3 |
| 9 | Tienda Piura | Rechazado | 1 |
| 10 | Tienda San Isidro | En Tránsito | 2 |
| 11 | Tienda Trujillo | En Preparación | 1 |
| 12 | Tienda Trujillo | Rechazado | 1 |

17. Comparación de ventas por método de pago

Esta consulta muestra el total de ventas realizadas a través de cada método de pago, lo que puede ser útil para identificar qué métodos de pago son más populares.

```
SELECT mp.nombre AS Metodo_Pago,  
       SUM(pa.importe) AS Total_Ventas  
FROM pagos pa  
JOIN metodo_pago mp ON pa.metodo_pago_id = mp.id  
GROUP BY mp.nombre  
ORDER BY Total_Ventas DESC;
```

| | Metodo_Pago | Total_Ventas |
|---|------------------------|--------------|
| 1 | Tarjeta de débito | 2369.10 |
| 2 | Tarjeta de crédito | 2259.10 |
| 3 | Banca por internet | 1969.11 |
| 4 | QR Yape/Plin | 1748.90 |
| 5 | PayPal | 1479.40 |
| 6 | Transferencia bancaria | 1289.30 |
| 7 | Bitcoin | 979.40 |

18. Análisis de evolución de ventas por mes y tiendas

Esta consulta muestra cómo han evolucionado las ventas mensuales de cada tienda. Es útil para realizar un análisis temporal y ver cómo las ventas fluctúan a lo largo del tiempo.

```
SELECT t.nombre AS Tienda,  
       YEAR(p.fecha_pedido) AS Anio,  
       MONTH(p.fecha_pedido) AS Mes,  
       SUM(p.total) AS Total_Ventas  
FROM pedidos p  
JOIN tiendas t ON p.tienda_id = t.id  
GROUP BY t.nombre, YEAR(p.fecha_pedido), MONTH(p.fecha_pedido)  
ORDER BY t.nombre, Anio DESC, Mes DESC;
```

| | Tienda | Anio | Mes | Total_Ventas |
|----|--------------------|------|-----|--------------|
| 1 | Tienda Arequipa | 2024 | 12 | 1349.40 |
| 2 | Tienda Barranco | 2024 | 12 | 1499.00 |
| 3 | Tienda Callao | 2024 | 12 | 889.60 |
| 4 | Tienda Chiclayo | 2024 | 12 | 879.60 |
| 5 | Tienda Cusco | 2024 | 12 | 1469.50 |
| 6 | Tienda Lima Centro | 2024 | 12 | 639.80 |
| 7 | Tienda Miraflores | 2024 | 12 | 1519.30 |
| 8 | Tienda Piura | 2024 | 12 | 1209.30 |
| 9 | Tienda San Isidro | 2024 | 12 | 1279.50 |
| 10 | Tienda Trujillo | 2024 | 12 | 1359.40 |

19. Clientes con mayor gasto promedio por pedido

Esta consulta identifica a los clientes que tienen un gasto promedio más alto por pedido. Es útil para identificar a los clientes más valiosos según el importe promedio que gastan en cada compra. Para este caso, el parámetro será clientes con un gasto mayor a 300.

```
SELECT c.id AS Cliente_ID,  
       c.nombre AS Cliente,  
       c.apellido_paterno,  
       AVG(p.total) AS Gasto_Promedio  
FROM clientes c  
JOIN pedidos p ON c.id = p.cliente_id  
GROUP BY c.id, c.nombre, c.apellido_paterno  
HAVING AVG(p.total) > 300  
ORDER BY Gasto_Promedio DESC;
```

| | Cliente_ID | Cliente | apellido_paterno | Gasto_Promedio |
|---|------------|---------|------------------|----------------|
| 1 | 14 | Juan | Aguirre | 459.900000 |
| 2 | 13 | Ricardo | García | 414.800000 |
| 3 | 7 | Carlos | Martínez | 389.900000 |
| 4 | 8 | Roberto | Vásquez | 359.900000 |
| 5 | 16 | Carlos | Quispe | 349.850000 |
| 6 | 2 | Carlos | Ramírez | 343.133333 |
| 7 | 12 | Marta | Cáceres | 334.850000 |
| 8 | 17 | Natalia | Vega | 329.900000 |
| 9 | 5 | Pedro | Gómez | 303.200000 |

20. Productos con el mayor número de pedidos y su disponibilidad actual de stock

Esta consulta obtiene los productos que han sido solicitados en la mayor cantidad de pedidos, junto con su cantidad disponible en stock en cada tienda. Esto ayuda a identificar productos populares que podrían necesitar reposición.

```
SELECT pr.nombre AS Producto,  
       COUNT(dp.pedido_id) AS Numero_Pedidos,  
       st.cantidad AS Stock_Disponible  
FROM detalles_pedido dp  
JOIN productos pr ON dp.producto_id = pr.id  
JOIN stock_tienda st ON pr.id = st.producto_id  
GROUP BY pr.nombre, st.cantidad  
ORDER BY Numero_Pedidos DESC;
```

| | Producto | Numero_Pedidos | Stock_Disponible |
|----|-----------------------------|----------------|------------------|
| 1 | Lenovo ThinkPad X1 | 40 | 5 |
| 2 | MacBook Pro 14" | 28 | 5 |
| 3 | Xiaomi Redmi Note 12 | 20 | 5 |
| 4 | iPhone 15 | 20 | 10 |
| 5 | AirPods Pro 2 | 16 | 10 |
| 6 | Xiaomi Redmi Note 12 | 16 | 10 |
| 7 | Samsung Galaxy Buds 2 | 15 | 20 |
| 8 | Samsung Galaxy Watch 5 | 12 | 10 |
| 9 | MacBook Pro 14" | 12 | 10 |
| 10 | Microsoft Surface Laptop 4 | 12 | 5 |
| 11 | iPhone 15 | 12 | 5 |
| 12 | Cargador inalámbrico Anker | 12 | 30 |
| 13 | Dell XPS 13 | 10 | 5 |
| 14 | Samsung Galaxy Buds 2 | 10 | 10 |
| 15 | Samsung Galaxy Buds 2 | 10 | 15 |
| 16 | Samsung Galaxy Watch 5 | 9 | 5 |
| 17 | Samsung Galaxy S23 | 8 | 10 |
| 18 | Funda OtterBox para iPho... | 8 | 10 |
| 19 | HP Pavilion 15 | 8 | 10 |
| 20 | Cargador inalámbrico Anker | 8 | 20 |
| 21 | AirPods Pro 2 | 8 | 15 |
| 22 | Cargador inalámbrico Anker | 8 | 40 |
| 23 | Cargador inalámbrico Anker | 8 | 50 |
| 24 | Samsung Galaxy Watch 5 | 6 | 20 |
| 25 | Dell XPS 13 | 6 | 10 |
| 26 | Microsoft Surface Laptop 4 | 6 | 10 |
| 27 | Samsung Galaxy Buds 2 | 5 | 50 |
| 28 | Samsung Galaxy Buds 2 | 5 | 40 |
| 29 | Samsung Galaxy Buds 2 | 5 | 30 |
| 30 | AirPods Pro 2 | 4 | 40 |

Las consultas propuestas sirven como modelo de una base para que las partes interesadas del negocio (*stakeholders*) puedan obtener una información clave sobre las operaciones y el rendimiento del negocio, así como para, posteriormente, evaluar tendencias de ventas a largo plazo. Con la capacidad de generar reportes y consultas como estas, se pueden tomar decisiones informadas y mejorar la eficiencia del sistema.

VII. LINKS DE INTERÉS

- Repositorio GitHub en donde se encontrará la información completa del trabajo (script, consultas, caso, gráficos): https://github.com/andrestume/dsrp_bootcampsql_e-commerce.git