Base de datos con SQL SERVER

Proyecto Final

AUTOR:

OSCAR CALSINA LAURA

Docente:

Ing. Kevin Rivera Vergaray

Lima, Perú

2024

Introducción

En el mundo actual, donde el comercio electrónico desempeña un papel crucial en el desarrollo económico, es esencial contar con sistemas eficientes para la gestión de información. Este proyecto consiste en el diseño e implementación de una base de datos para una tienda en línea, que permita registrar y gestionar datos relacionados con productos, clientes, pedidos y otros aspectos fundamentales del negocio. La base de datos ha sido desarrollada utilizando Microsoft SQL Server y busca optimizar la operación de una tienda ficticia con procesos automatizados y consultas eficientes.

Descripción del Caso

El caso se centra en una tienda en línea que ofrece una amplia variedad de productos tecnológicos organizados en categorías y marcas. Los clientes realizan pedidos que incluyen múltiples productos y pueden pagar mediante diferentes métodos. Adicionalmente, los pedidos son enviados a direcciones específicas, y se registran detalles de los envíos, así como los pagos realizados.

La base de datos tiene como objetivo proporcionar una solución estructurada para manejar los datos generados por estas operaciones y facilitar la toma de decisiones a través de consultas e informes.

Descripción del Problema

Antes de la implementación de esta base de datos, los datos de la tienda se gestionaban de manera desorganizada, lo que dificultaba:

- El seguimiento de los pedidos y su estado.
- La gestión del inventario de productos.
- La generación de informes sobre ventas y clientes.
- La automatización de procesos como la actualización de stock o el registro de pagos.

Objetivo

El objetivo principal es diseñar e implementar una base de datos relacional que permita:

1. Organizar la información de la tienda de manera eficiente.
2. Automatizar procesos clave como la actualización de inventario.
3. Facilitar consultas complejas para obtener información relevante sobre clientes, pedidos, pagos y productos.
4. Proveer una base sólida para el análisis de datos y la generación de informes.

Descripción de las Tablas y Relaciones

Tablas Principales

1. **Categorías**
    - **ID_categoría**: INT, Primary Key.
    - **Nombre**: VARCHAR(50).
2. **Marcas**
    - **ID_marca**: INT, Primary Key.
    - **Nombre**: VARCHAR(50).
3. **Tipos_clientes**
    - **ID_tipo_cliente**: INT, Primary Key.
    - **Descripción**: VARCHAR(50).
4. **Métodos_pago**
    - **ID_método_pago**: INT, Primary Key.
    - **Descripción**: VARCHAR(50).
5. **Productos**
    - **ID_producto**: INT, Primary Key.
    - **Nombre**: VARCHAR(100), NOT NULL.
    - **Descripción**: TEXT.
    - **Precio**: DECIMAL(10,2).
    - **Stock**: INT.
    - **Categoría_id**: INT, Foreign Key (Referencias Categorías.ID_categoría).
    - **Marca_id**: INT, Foreign Key (Referencias Marcas.ID_marca).
6. **Clientes**
    - **ID_cliente**: INT, Primary Key.
    - **Nombre**: VARCHAR(50).
    - **Apellido**: VARCHAR(50).
    - **Correo_electronico**: VARCHAR(100).
    - **Tipo_cliente_id**: INT, Foreign Key (Referencias Tipos_clientes.ID_tipo_cliente).
    -

7. **Pedidos**
    - **ID_pedido**: INT, Primary Key.
    - **Cliente_id**: INT, Foreign Key (Referencias Clientes.ID_cliente).
    - **Fecha_pedido**: DATETIME.
    - **Estado**: VARCHAR(50).
    - **Total**: DECIMAL(10,2).
8. **Detalles_pedido**
    - **ID_detalle_pedido**: INT, Primary Key.
    - **Pedido_id**: INT, Foreign Key (Referencias Pedidos.ID_pedido).
    - **Producto_id**: INT, Foreign Key (Referencias Productos.ID_producto).
    - **Cantidad**: INT.
    - **Precio_unitario**: DECIMAL(10,2).
9. **Pagos**
    - **ID_pago**: INT, Primary Key.
    - **Pedido_id**: INT, Foreign Key (Referencias Pedidos.ID_pedido).
    - **Método_pago_id**: INT, Foreign Key (Referencias Métodos_pago.ID_método_pago).
    - **Monto**: DECIMAL(10,2).
    - **Fecha_pago**: DATETIME.
10. **Direcciones**
    - **ID_dirección**: INT, Primary Key.
    - **Cliente_id**: INT, Foreign Key (Referencias Clientes.ID_cliente).
    - **Tipo_dirección**: VARCHAR(50).
    - **Calle**: VARCHAR(100).
    - **Número**: VARCHAR(20).
    - **Ciudad**: VARCHAR(50).
    - **Código_postal**: VARCHAR(20).
    - **País**: VARCHAR(50).

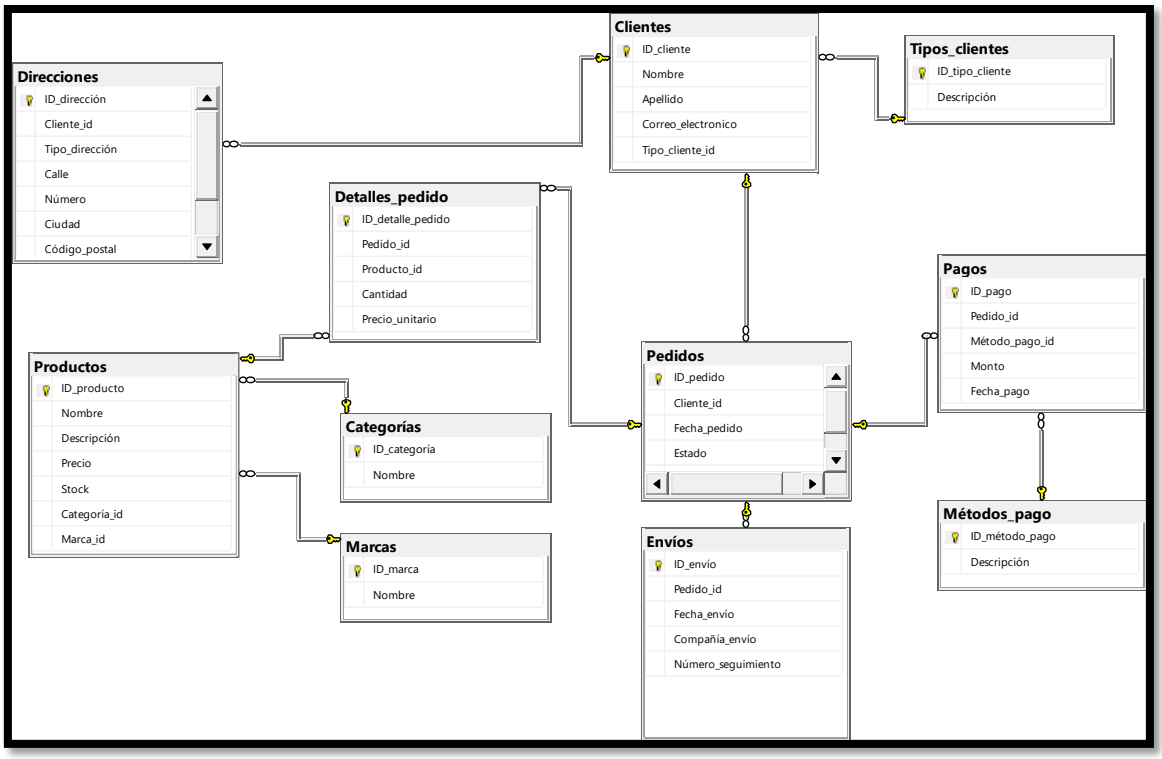
11. **Envíos**

- **ID_envío**: INT, Primary Key.
- **Pedido_id**: INT, Foreign Key (Referencias Pedidos.ID_pedido).
- **Fecha_envío**: DATETIME.
- **Compañía_envío**: VARCHAR(50).
- **Número_seguimiento**: VARCHAR(50).

Relaciones

- **Clientes** se relaciona con **Direcciones** (1:N) y **Pedidos** (1:N).
- **Pedidos** se relaciona con **Detalles_pedido** (1:N), **Pagos** (1:1), y **Envíos** (1:1).
- **Productos** se relaciona con **Detalles_pedido** (1:N) y tiene relaciones con **Categorías** (N:1) y **Marcas** (N:1).
- **Métodos_pago** se relaciona con **Pagos** (1:N).
- **Tipos_clientes** se relaciona con **Clientes** (1:N).

Diseño Físico



Anexos

Creación de base de datos y tablas

```
CREATE DATABASE proyectosql;
USE proyectosql;

-- Tablas de referencia
CREATE TABLE Categorías (
   ID_categoría INT IDENTITY(1,1) PRIMARY KEY,
   Nombre VARCHAR(50)
);

CREATE TABLE Marcas (
   ID_marca INT IDENTITY(1,1) PRIMARY KEY,
   Nombre VARCHAR(50)
);

CREATE TABLE Tipos_clientes (
   ID_tipo_cliente INT IDENTITY(1,1) PRIMARY KEY,
   Descripción VARCHAR(50)
);

CREATE TABLE Métodos_pago (
   ID_método_pago INT IDENTITY(1,1) PRIMARY KEY,
   Descripción VARCHAR(50)
);

-- Tablas principales
CREATE TABLE Productos (
```

```
    ID_producto INT IDENTITY(1,1) PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL,
    Descripción TEXT,
    Precio DECIMAL(10,2),
    Stock INT,
    Categoría_id INT FOREIGN KEY REFERENCES Categorías(ID_categoría),
    Marca_id INT FOREIGN KEY REFERENCES Marcas(ID_marca)
);

CREATE TABLE Clientes (
    ID_cliente INT IDENTITY(1,1) PRIMARY KEY,
    Nombre VARCHAR(50),
    Apellido VARCHAR(50),
    Correo_electronico VARCHAR(100),
    Tipo_cliente_id INT FOREIGN KEY REFERENCES Tipos_clientes(ID_tipo_cliente)
);

CREATE TABLE Pedidos (
    ID_pedido INT IDENTITY(1,1) PRIMARY KEY,
    Cliente_id INT FOREIGN KEY REFERENCES Clientes(ID_cliente),
    Fecha_pedido DATETIME,
    Estado VARCHAR(50),
    Total DECIMAL(10,2)
);

CREATE TABLE Detalles_pedido (
    ID_detalle_pedido INT IDENTITY(1,1) PRIMARY KEY,
    Pedido_id INT FOREIGN KEY REFERENCES Pedidos(ID_pedido),
    Producto_id INT FOREIGN KEY REFERENCES Productos(ID_producto),
```

```
    Cantidad INT,
    Precio_unitario DECIMAL(10,2)
);

CREATE TABLE Pagos (
    ID_pago INT IDENTITY(1,1) PRIMARY KEY,
    Pedido_id INT FOREIGN KEY REFERENCES Pedidos(ID_pedido),
    Método_pago_id INT FOREIGN KEY REFERENCES Métodos_pago(ID_método_pago),
    Monto DECIMAL(10,2),
    Fecha_pago DATETIME
);

CREATE TABLE Direcciones (
    ID_dirección INT IDENTITY(1,1) PRIMARY KEY,
    Cliente_id INT FOREIGN KEY REFERENCES Clientes(ID_cliente),
    Tipo_dirección VARCHAR(50),
    Calle VARCHAR(100),
    Número VARCHAR(20),
    Ciudad VARCHAR(50),
    Código_postal VARCHAR(20),
    País VARCHAR(50)
);

CREATE TABLE Envíos (
    ID_envío INT IDENTITY(1,1) PRIMARY KEY,
    Pedido_id INT FOREIGN KEY REFERENCES Pedidos(ID_pedido),
    Fecha_envío DATETIME,
    Compañía_envío VARCHAR(50),
    Número_seguimiento VARCHAR(50)
```

);

Inserción de datos

```
USE proyectosql;
-- Insertar datos en las tablas de referencia
INSERT INTO Categorías (Nombre)
VALUES
    ('Electrónica'),
    ('Ropa'),
    ('Libros'),
    ('Deportes'),
    ('Belleza');


USE proyectosql;

INSERT INTO Marcas (Nombre)
VALUES
    ('Samsung'),
    ('Apple'),
    ('Nike'),
    ('Adidas'),
    ('Sony'),
    ('LG'),
    ('H&M'),
    ('Zara'),
    ('Penguin'),
    ('Under Armour');
```

```
INSERT INTO Tipos_clientes (Descripción)
VALUES
    ('Mayorista'),
    ('Minorista'),
    ('Corporativo'),
    ('Individual');

INSERT INTO Métodos_pago (Descripción)
VALUES
    ('Tarjeta de crédito'),
    ('Transferencia bancaria'),
    ('Efectivo'),
    ('PayPal');

-- Insertar datos en las tablas principales
INSERT INTO Productos (Nombre, Descripción, Precio, Stock, Categoría_id, Marca_id)
VALUES
    ('Smartphone Galaxy S23', 'Smartphone de alta gama con pantalla AMOLED', 1200.99, 30, 1, 1),
    ('iPhone 14 Pro Max', 'Smartphone premium con cámara Pro', 1500.99, 25, 1, 2),
    ('Nike Air Max', 'Zapatillas deportivas de running', 129.99, 50, 3, 3),
    ('Adidas Ultraboost', 'Zapatillas de running de alta performance', 159.99, 40, 3, 4),
    ('Sony PlayStation 5', 'Consola de videojuegos de última generación', 549.99, 15, 1, 5),
    ('LG OLED TV', 'Televisor OLED de alta resolución', 1899.99, 10, 1, 6),
    ('Camiseta Nike', 'Camiseta deportiva de algodón', 29.99, 100, 2, 3),
    ('Pantalón Jeans Levis', 'Pantalón jeans clásico', 79.99, 80, 2, 7),
    ('Libro El Señor de los Anillos', 'Trilogía épica de fantasía', 29.99, 20, 3, 9),
    ('Pelota de fútbol', 'Pelota de fútbol oficial', 19.99, 50, 3, 4),
    ('Crema facial', 'Crema hidratante para piel seca', 25.99, 40, 5, 10),
    ('Maquillaje', 'Paleta de sombras de ojos', 39.99, 30, 5, 10),
```

```
    ('Perfume', 'Perfume floral para mujer', 59.99, 20, 5, 10),
    ('Bicicleta de montaña', 'Bicicleta de montaña de aluminio', 499.99, 10, 3, 4),
    ('Raqueta de tenis', 'Raqueta de tenis profesional', 129.99, 15, 3, 3);

INSERT INTO Clientes (Nombre, Apellido, Correo_electronico, Tipo_cliente_id)
VALUES
    ('Juan', 'Pérez', 'juanperez@gmail.com', 1),
    ('María', 'López', 'marialopez@hotmail.com', 2),
    ('Carlos', 'García', 'carlosgarcia@outlook.com', 3),
    ('Ana', 'Rodríguez', 'anarodriguez@yahoo.com', 4),
    ('Pedro', 'Martínez', 'pedromartinez@gmail.com', 1),
    ('Laura', 'Hernández', 'laurahernandez@hotmail.com', 2),
    ('David', 'González', 'davidgonzalez@outlook.com', 3),
    ('Sofía', 'Fernández', 'sofiafernandez@yahoo.com', 4),
    ('Diego', 'Díaz', 'diegodíaz@gmail.com', 1),
    ('Andrea', 'Sánchez', 'andreasanchez@hotmail.com', 2),
    ('Pablo', 'Moreno', 'pablomoreno@outlook.com', 3),
    ('Valentina', 'Jiménez', 'valentinajimenez@yahoo.com', 4),
    ('Lucas', 'Romero', 'lucasromero@gmail.com', 1),
    ('Camila', 'Vidal', 'camilavidal@hotmail.com', 2),
    ('Mateo', 'Castro', 'mateocastro@outlook.com', 3);

INSERT INTO Pedidos (Cliente_id, Fecha_pedido, Estado, Total)
VALUES
    (1, '2023-11-20', 'Completado', 1200.99),
    (2, '2023-11-15', 'Procesando', 500.49),
    (3, '2023-11-22', 'Entregado', 299.99),
    (4, '2023-11-18', 'Cancelado', 159.99),
    (5, '2023-11-21', 'Completado', 549.99),
```

(6, '2023-11-19', 'Procesando', 1899.99),

(7, '2023-11-23', 'Entregado', 29.99),

(8, '2023-11-17', 'Cancelado', 79.99),

(9, '2023-11-24', 'Completado', 29.99),

(10, '2023-11-20', 'Procesando', 19.99),

(11, '2023-11-22', 'Entregado', 25.99),

(12, '2023-11-18', 'Cancelado', 39.99),

(13, '2023-11-21', 'Completado', 59.99),

(14, '2023-11-19', 'Procesando', 499.99),

(15, '2023-11-23', 'Entregado', 129.99);

INSERT INTO Direcciones (Cliente_id, Tipo_dirección, Calle, Número, Ciudad, Código_postal, País)

VALUES

(1, 'Residencial', 'Avenida Principal', '123', 'Ciudad Ejemplo', '12345', 'País Ejemplo'),

(2, 'Comercial', 'Calle Secundaria', '456', 'Pueblo Ejemplo', '67890', 'País Ejemplo'),

(3, 'Residencial', 'Calle de la Paz', '789', 'Villa Alegre', '54321', 'País Ejemplo');

INSERT INTO Pagos (Pedido_id, Método_pago_id, Monto, Fecha_pago)

VALUES

(1, 1, 120.50, '2023-11-25 13:30:00'),

(2, 2, 50.00, '2023-11-22 10:15:00'),

(3, 1, 250.99, '2023-12-01 16:45:00');

INSERT INTO Envíos (Pedido_id, Fecha_envío, Compañía_envío, Número_seguimiento)

VALUES

(1, '2023-11-27', 'DHL', '1Z2Y3X4W5V'),

```
    (2, '2023-11-23', 'FedEx', 'ABCDEF12345'),
    (3, '2023-12-02', 'UPS', '987654321Z');

INSERT INTO Detalles_pedido (Pedido_id, Producto_id, Cantidad, Precio_unitario)
VALUES
    (1, 1, 2, 19.99),
    (1, 3, 1, 29.99),
    (2, 2, 3, 15.99),
    (3, 1, 1, 19.99),
    (3, 4, 2, 39.99);
```

Consultas

1.- -- Consulta para obtener el producto más solicitado

```
SELECT TOP 1
    P.Nombre AS Producto,
    SUM(DP.Cantidad) AS Total_Cantidad_Pedida
FROM
    Detalles_pedido DP
INNER JOIN
    Productos P ON DP.Producto_id = P.ID_producto
GROUP BY
    P.Nombre
ORDER BY
    Total_Cantidad_Pedida DESC;
```

2.- -- Consulta para obtener los datos de clientes, sus direcciones y los pedidos realizados

```
SELECT
    C.Nombre AS Nombre,
```

```
    C.Apellido AS Apellido,
    D.Tipo_dirección AS Tipo_Dirección,
    D.Calle AS Calle,
    D.Número AS Número,
    D.Ciudad AS Ciudad,
    D.Código_postal AS Código_Postal,
    D.País AS País,
    P.ID_pedido AS Pedido_ID,
    P.Fecha_pedido AS Fecha_Pedido,
    P.Estado AS Estado_Pedido,
    P.Total AS Total_Pedido
FROM
    Clientes C
INNER JOIN
    Direcciones D ON C.ID_cliente = D.Cliente_id
INNER JOIN
    Pedidos P ON C.ID_cliente = P.Cliente_id;
```

3.- -- Consulta para obtener los clientes con pedidos en estado "Completado"

```
SELECT
    C.Nombre AS Nombre,
    C.Apellido AS Apellido,
    P.Estado AS Estado_Pedido
FROM
    Clientes C
INNER JOIN
    Pedidos P ON C.ID_cliente = P.Cliente_id
WHERE
    P.Estado = 'Completado';
```

4.- --Crea un procedimiento almacenado para consultar los pedidos de acuerdo con un estado específico.

```
CREATE PROCEDURE ObtenerPedidosPorEstado
    @EstadoPedido VARCHAR(50)
AS
BEGIN
    SELECT
        P.ID_pedido AS Pedido_ID,
        P.Fecha_pedido AS Fecha_Pedido,
        P.Estado AS Estado_Pedido,
        P.Total AS Total_Pedido,
        C.Nombre AS Cliente_Nombre,
        C.Apellido AS Cliente_Apellido
    FROM
        Pedidos P
    INNER JOIN
        Clientes C ON P.Cliente_id = C.ID_cliente
    WHERE
        P.Estado = @EstadoPedido;
END;

EXEC ObtenerPedidosPorEstado 'Completado';

EXEC ObtenerPedidosPorEstado 'Procesando';
```

5.- --Cambia el estado de un pedido existente.

```
CREATE PROCEDURE ActualizarEstadoPedido
    @PedidoID INT,
    @NuevoEstado VARCHAR(50)
AS
BEGIN
    UPDATE Pedidos
    SET Estado = @NuevoEstado
    WHERE ID_pedido = @PedidoID;
END;

--ejecutando
EXEC ActualizarEstadoPedido
    @PedidoID = 2,
    @NuevoEstado = 'Completado';
```

6.- --Lista los productos de una categoría específica.

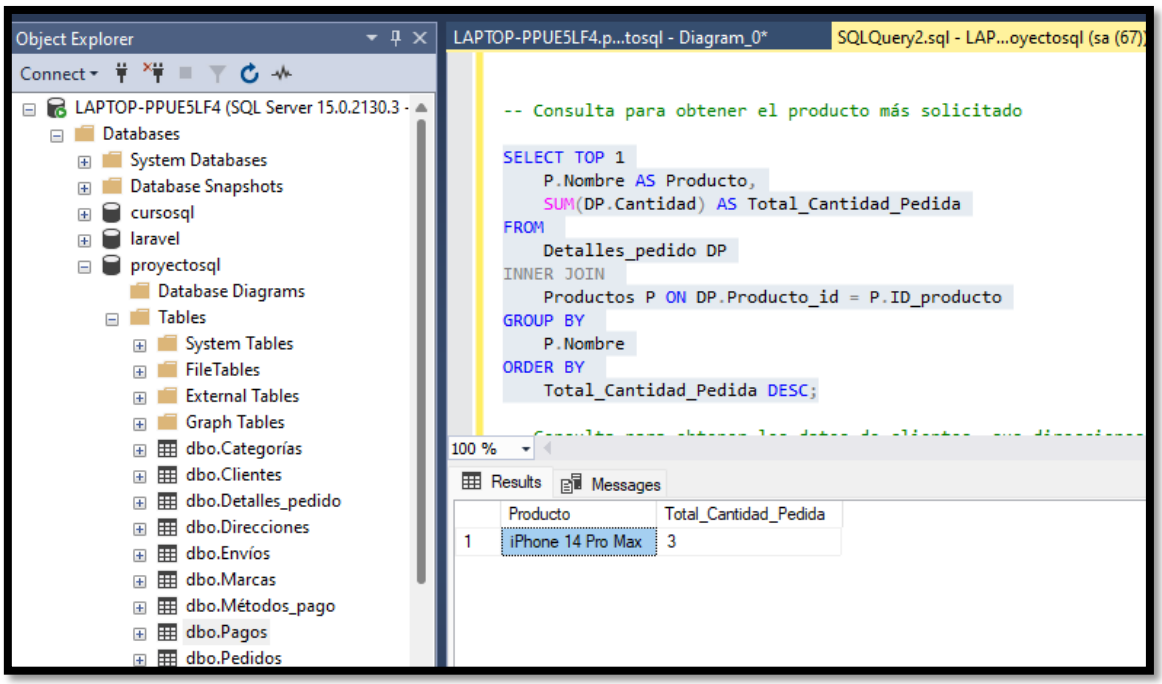
```
CREATE PROCEDURE BuscarProductosPorCategoria
    @CategoriaNombre VARCHAR(50)
AS
BEGIN
    SELECT
        P.ID_producto AS Producto_ID,
        P.Nombre AS Nombre_Producto,
        P.Descripción AS Descripción,
        P.Precio AS Precio,
        P.Stock AS Stock,
        C.Nombre AS Nombre_Categoría
    FROM
```

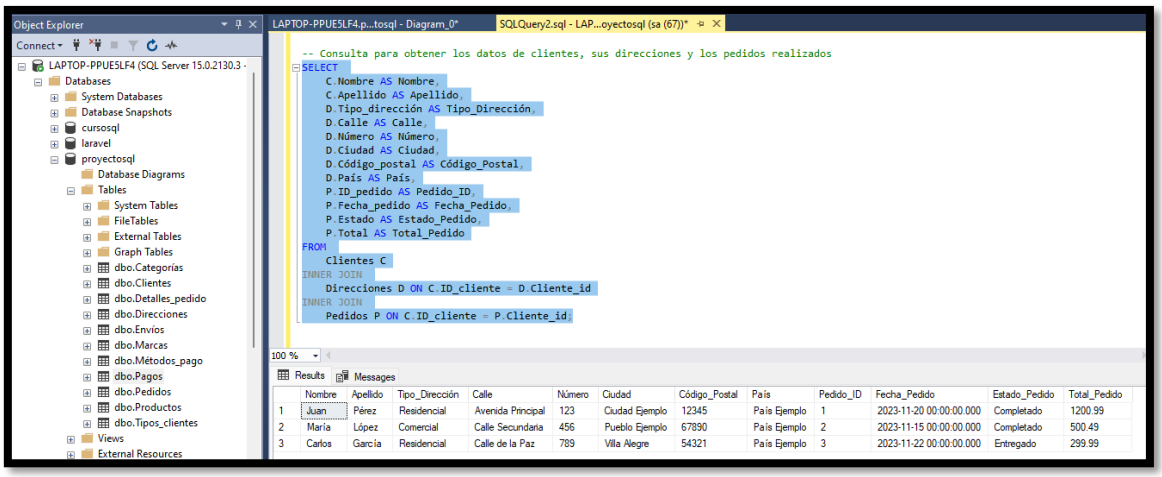
```
    Productos P
  INNER JOIN
    Categorías C ON P.Categoría_id = C.ID_categoría
  WHERE
    C.Nombre = @CategoriaNombre;
END;
```

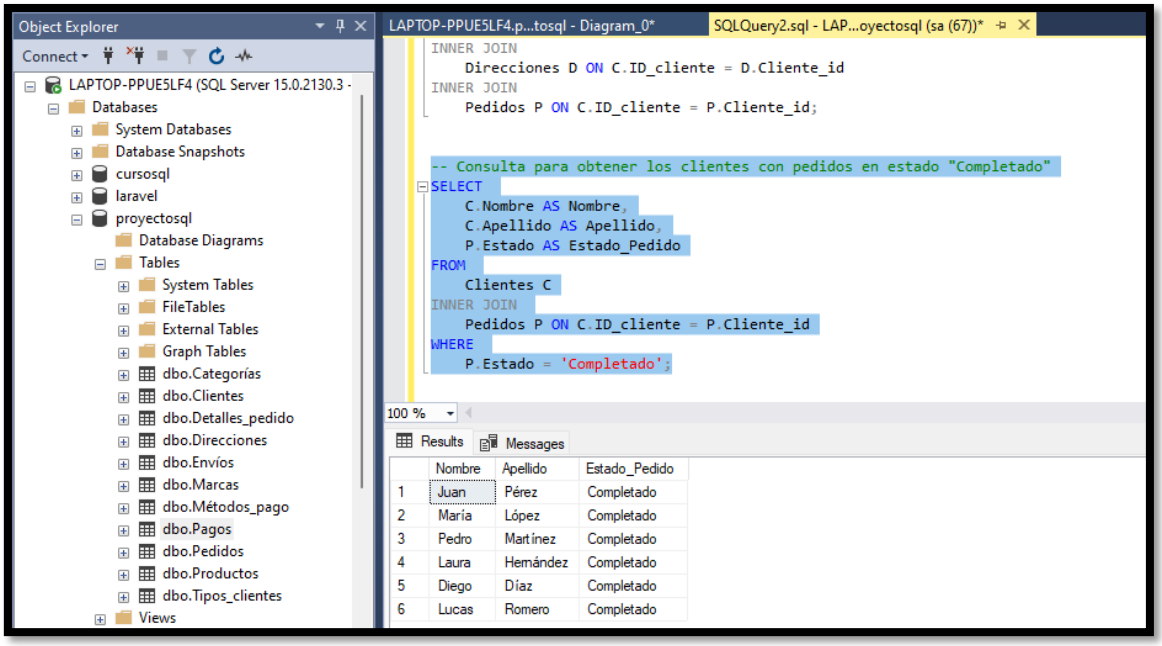
7.- -- Obtén un resumen de ventas dentro de un rango de fechas.

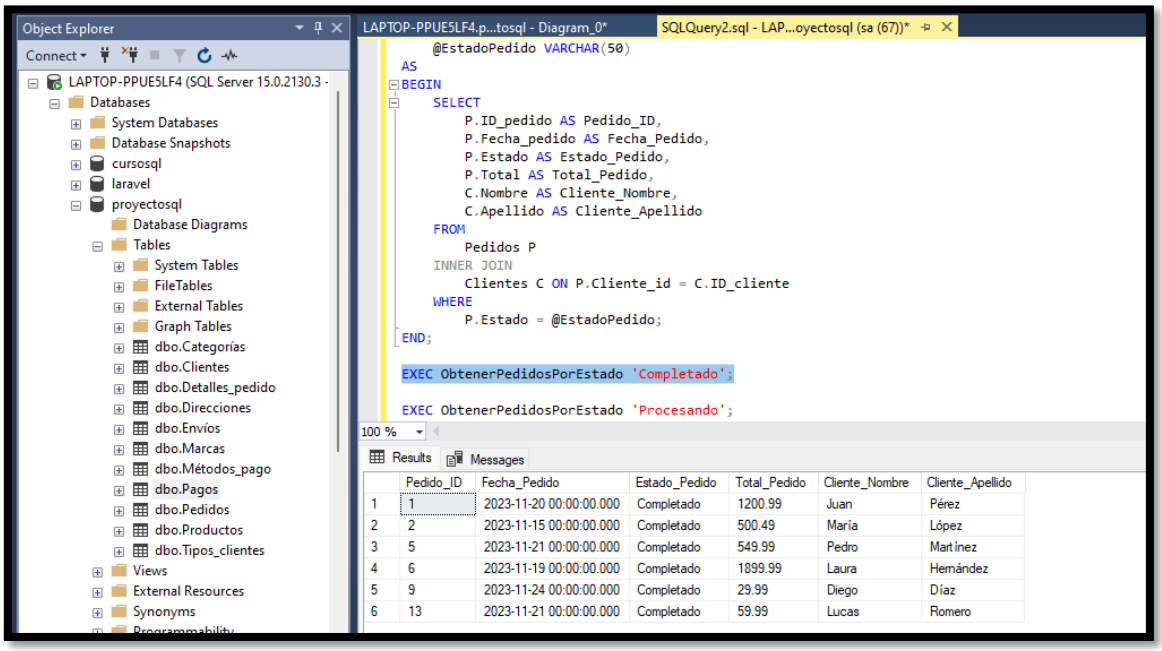
```
CREATE PROCEDURE InformeVentas
  @FechaInicio DATETIME,
  @FechaFin DATETIME
AS
BEGIN
  SELECT
    P.ID_pedido AS Pedido_ID,
    P.Fecha_pedido AS Fecha_Pedido,
    P.Total AS Total_Pedido,
    C.Nombre AS Cliente_Nombre,
    C.Apellido AS Cliente_Apellido
  FROM
    Pedidos P
  INNER JOIN
    Clientes C ON P.Cliente_id = C.ID_cliente
  WHERE
    P.Fecha_pedido BETWEEN @FechaInicio AND @FechaFin;
END;
```

Resultados:


Object Explorer
Connect
LAPTOP-PPUE5LF4 (SQL Server 15.0.2130.3 -
Databases
System Databases
Database Snapshots
cursosql
laravel
proyectosql
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Categorías
dbo.Clientes
dbo.Detalles_pedido
dbo.Direcciones
dbo.Envíos
dbo.Marcas
dbo.Métodos_pago
dbo.Pagos
dbo.Pedidos
LAPTOP-PPUE5LF4.p...tosql - Diagram_0*
SQLQuery2.sql - LAP...oyectosql (sa (67))
-- Consulta para obtener el producto más solicitado
SELECT TOP 1
P.Nombre AS Producto,
SUM(DP.Cantidad) AS Total_Cantidad_Pedida
FROM
Detalles_pedido DP
INNER JOIN
Productos P ON DP.Producto_id = P.ID_producto
GROUP BY
P.Nombre
ORDER BY
Total_Cantidad_Pedida DESC;
100 %
Results
Messages
Producto
Total_Cantidad_Pedida
1
iPhone 14 Pro Max
3


-- Consulta para obtener los datos de clientes, sus direcciones y los pedidos realizados
SELECT
C.Nombre AS Nombre,
C.Apellido AS Apellido,
D.Tipo_dirección AS Tipo_Dirección,
D.Calle AS Calle,
D.Número AS Número,
D.Ciudad AS Ciudad,
D.Código_postal AS Código_Postal,
D.País AS País,
P.ID_pedido AS Pedido_ID,
P.Fecha_pedido AS Fecha_Pedido,
P.Estado AS Estado_Pedido,
P.Total AS Total_Pedido
FROM
Clientes C
INNER JOIN
Direcciones D ON C.ID_cliente = D.Cliente_id
INNER JOIN
Pedidos P ON C.ID_cliente = P.Cliente_id;
Nombre Apellido Tipo_Dirección Calle Número Ciudad Código_Postal País Pedido_ID Fecha_Pedido Estado_Pedido Total_Pedido
1 Juan Pérez Residencial Avenida Principal 123 Ciudad Ejemplo 12345 País Ejemplo 1 2023-11-20 00:00:00.000 Completado 1200.99
2 María López Comercial Calle Secundaria 456 Pueblo Ejemplo 67890 País Ejemplo 2 2023-11-15 00:00:00.000 Completado 500.49
3 Carlos García Residencial Calle de la Paz 789 Villa Alegre 54321 País Ejemplo 3 2023-11-22 00:00:00.000 Entregado 299.99


Object Explorer
Connect
LAPTOP-PPUE5LF4 (SQL Server 15.0.2130.3 -
Databases
System Databases
Database Snapshots
cursosql
laravel
proyectosql
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Categorías
dbo.Clientes
dbo.Detalles_pedido
dbo.Direcciones
dbo.Envíos
dbo.Marcas
dbo.Métodos_pago
dbo.Pagos
dbo.Pedidos
dbo.Productos
dbo.Tipos_clientes
Views
LAPTOP-PPUE5LF4.p...tosql - Diagram_0*
SQLQuery2.sql - LAP...oyectosql (sa (67))*
INNER JOIN
Direcciones D ON C.ID_cliente = D.Cliente_id
INNER JOIN
Pedidos P ON C.ID_cliente = P.Cliente_id;
-- Consulta para obtener los clientes con pedidos en estado "Completado"
SELECT
C.Nombre AS Nombre,
C.Apellido AS Apellido,
P.Estado AS Estado_Pedido
FROM
Clientes C
INNER JOIN
Pedidos P ON C.ID_cliente = P.Cliente_id
WHERE
P.Estado = 'Completado';
100 %
Results
Messages
Nombre Apellido Estado_Pedido
1 Juan Pérez Completado
2 María López Completado
3 Pedro Martínez Completado
4 Laura Hernández Completado
5 Diego Díaz Completado
6 Lucas Romero Completado


Object Explorer
Connect
LAPTOP-PPUE5LF4 (SQL Server 15.0.2130.3 -
Databases
System Databases
Database Snapshots
cursosql
laravel
proyectosql
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Categorías
dbo.Clientes
dbo.Detalles_pedido
dbo.Direcciones
dbo.Envíos
dbo.Marcas
dbo.Métodos_pago
dbo.Pagos
dbo.Pedidos
dbo.Productos
dbo.Tipos_clientes
Views
External Resources
Synonyms
LAPTOP-PPUE5LF4.p...tosql - Diagram_0*
SQLQuery2.sql - LAP...oyectosql (sa (67))*
@EstadoPedido VARCHAR(50)
AS
BEGIN
SELECT
P.ID_pedido AS Pedido_ID,
P.Fecha_pedido AS Fecha_Pedido,
P.Estado AS Estado_Pedido,
P.Total AS Total_Pedido,
C.Nombre AS Cliente_Nombre,
C.Apellido AS Cliente_Apellido
FROM
Pedidos P
INNER JOIN
Clientes C ON P.Cliente_id = C.ID_cliente
WHERE
P.Estado = @EstadoPedido;
END;
EXEC ObtenerPedidosPorEstado 'Completado';
EXEC ObtenerPedidosPorEstado 'Procesando';
100 %
Results
Messages
Pedido_ID Fecha_Pedido Estado_Pedido Total_Pedido Cliente_Nombre Cliente_Apellido
1 1 2023-11-20 00:00:00.000 Completado 1200.99 Juan Pérez
2 2 2023-11-15 00:00:00.000 Completado 500.49 María López
3 5 2023-11-21 00:00:00.000 Completado 549.99 Pedro Martínez
4 6 2023-11-19 00:00:00.000 Completado 1899.99 Laura Hernández
5 9 2023-11-24 00:00:00.000 Completado 29.99 Diego Díaz
6 13 2023-11-21 00:00:00.000 Completado 59.99 Lucas Romero