

Documentación del Proyecto: Análisis de Ventas y Clientes

Proyecto Final de la materia Base de Datos de la Maestría en Minería de Datos de la UTN Rosario

Docente: Layla Scheli

Alumno: Federico Colombo

Fecha: 21/08/2024

1. Introducción

Este proyecto tiene como objetivo diseñar y construir una base de datos para analizar las ventas y el comportamiento de los clientes de la empresa ficticia "DeRemate". A través de este proyecto, se busca obtener información valiosa sobre las ventas y los clientes para mejorar la toma de decisiones y la estrategia comercial de la empresa.

2. Diseño de la Base de Datos

2.1. Esquema de la Base de Datos

La base de datos DeRemate se compone de tres tablas principales:

- Customers: Contiene la información de los clientes que han realizado compras.
- Products: Almacena los detalles de los productos disponibles para la venta.
- Sales: Registra las transacciones de ventas realizadas por los clientes.

2.2. Relación entre las Tablas

Customers y Sales:

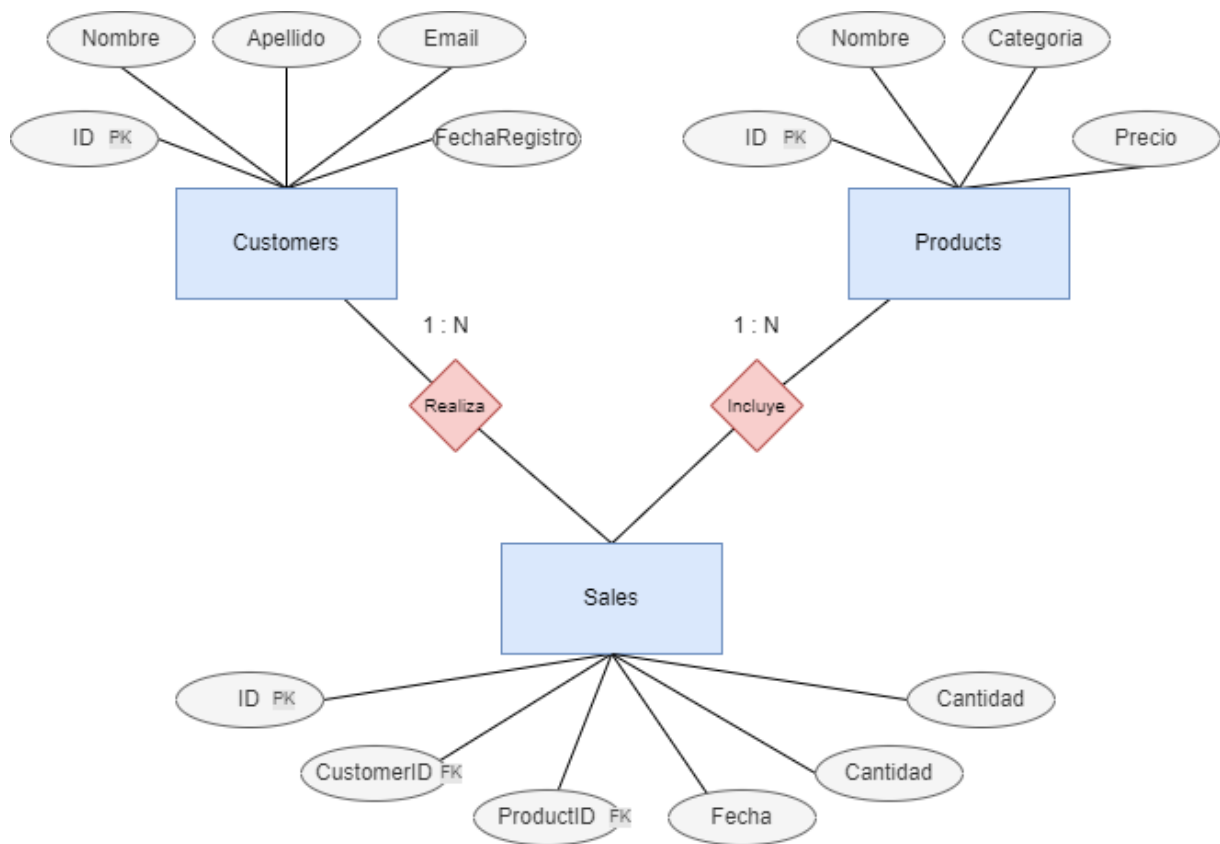
- Relación uno a muchos. Un cliente puede realizar múltiples compras, pero cada compra está asociada a un solo cliente.
- La clave primaria en Customers (ID) se relaciona con la clave foránea CustomerID en la tabla Sales.

Products y Sales:

- Relación uno a muchos. Un producto puede ser parte de múltiples ventas, pero cada venta está asociada a un solo producto.
- La clave primaria en Products (ID) se relaciona con la clave foránea ProductID en la tabla Sales.

2.3. Diagrama Entidad Relación

El siguiente diagrama ER muestra las relaciones entre las tablas Customers, Products, y Sales:



2.4. Documentación

El directorio 'documentacion/' incluye el diagrama ER.

3. Creación de la Base de Datos

Se han creado las siguientes tablas y atributos:

- Customers (ID, Nombre, Apellido, Email, Fecha de Registro)
- Products (ID, Nombre, Categoría, Precio)
- Sales (ID, CustomerID, ProductID, Fecha, Cantidad, Total)

Script SQL

El Script SQL utilizado para la creación del esquema de la base de datos se puede consultar en el Anexo 1. El directorio 'scripts/' contiene el Script SQL utilizado para crear la base de datos y las tablas.

4. Población de la Base de Datos

Se han insertado registros ficticios en cada una de las tablas para simular un entorno de ventas real. La tabla Customers contiene 10 registros de clientes, la tabla Products tiene 20 productos, y la tabla Sales contiene 30 transacciones de ventas.

	ID	Nombre	Apellido	Email	FechaRegistro
1	1	Juan	Perez	juan.perez@gmail.com	2023-01-15
2	2	Ana	Gomez	ana.gomez@gmail.com	2023-02-20
3	3	Luis	Martinez	luis.martinez@outlook.com	2023-03-10
4	4	Maria	Lopez	maria.lopez@gmail.com	2023-04-05
5	5	Carlos	Fernandez	carlos.fernandez@hotmail.com	2023-05-22
6	6	Sofia	Ramirez	sofia.ramirez@outlook.com	2023-06-15
7	7	Miguel	Suarez	miguel.suarez@fibertel.com	2023-07-01
8	8	Laura	Gonzalez	laura.gonzalez@argentina.com	2023-07-20

	ID	Nombre	Categoría	Precio
1	1	Laptop	Electronica	1200.00
2	2	Smartphone	Electronica	800.00
3	3	Tablet	Electronica	500.00
4	4	Monitor	Electronica	300.00
5	5	Teclado	Accesorios	50.00
6	6	Mouse	Accesorios	30.00
7	7	Impresora	Electronica	200.00
8	8	Camara	Fotografia	450.00

	ID	CustomerID	ProductID	Fecha	Cantidad	Total
1	1	1	1	2023-06-01	1	1200.00
2	2	1	5	2023-06-01	2	100.00
3	3	2	2	2023-06-05	1	800.00
4	4	2	6	2023-06-05	1	30.00
5	5	3	3	2023-06-10	1	500.00
6	6	3	7	2023-06-12	1	200.00
7	7	4	4	2023-06-15	2	600.00
8	8	4	9	2023-06-15	1	100.00
9	9	5	8	2023-06-20	1	450.00
10	10	5	10	2023-06-22	1	250.00

Scripts SQL

El Script SQL utilizado para la inserción de datos se puede consultar en el Anexo 2. El directorio 'scripts/' contiene el Script SQL utilizado para poblar la base de datos.

5. Consultas de Análisis

Se realizaron diversas consultas SQL para extraer información valiosa sobre las ventas y los clientes.

Los 5 productos más vendidos:

	NombreProducto	TotalVendido
1	Monitor	3
2	Ventilador	3
3	Cafetera	2
4	Auriculares	2
5	Laptop	2

Los 5 clientes que han gastado más en total:

	Nombre	Apellido	GastoTotal
1	Carlos	Fernandez	2100.00
2	Juan	Perez	1880.00
3	Maria	Lopez	1600.00
4	Javier	Rodriguez	1500.00
5	Laura	Gonzalez	1400.00

Ingreso total por categoría de producto:

	Categoria	IngresoTotal
1	Electronica	7600.00
2	Electrodomesticos	3780.00
3	Accesorios	780.00
4	Fotografia	450.00

Promedio de ventas diarias:

	PromedioVentasDiarias
1	525.416666

Script SQL

El Script SQL utilizado para las consultas de análisis se puede consultar en el Anexo 3. El directorio 'scripts/' contiene el Script SQL utilizado para las consultas de análisis.

5. Procedimientos Almacenados

Se crearon dos procedimientos almacenados para facilitar consultas específicas.

Obtener compras por cliente:

Ejemplo: Cliente con ID 1 (Juan Perez)

	IDVenta	NombreProducto	Fecha	Cantidad	Total
1	1	Laptop	2023-06-01	1	1200.00
2	2	Teclado	2023-06-01	2	100.00
3	11	Tablet	2023-07-01	1	500.00
4	26	Licuada	2023-08-10	1	80.00

Obtener ventas por rango de fechas:

Ejemplo: del 01-06-2023 al 01-07-2023

	VentasTotales
1	4730.00

Script SQL

El Script SQL utilizado para los procedimientos almacenados se puede consultar en el Anexo 4.

El directorio 'scripts/' contiene el Script SQL utilizado para los procedimientos almacenados.

6. Integración con Power BI

La base de datos DeRemate fue conectada a Power BI para crear visualizaciones. Se cargaron las tablas Customers, Products, y Sales con el fin de crear un Dashboard para que facilite la toma decisiones del negocio.

DeRemate

Análisis de Ventas y Clientes

12,61 mil

Ventas Totales \$

30

Cantidad de Ventas

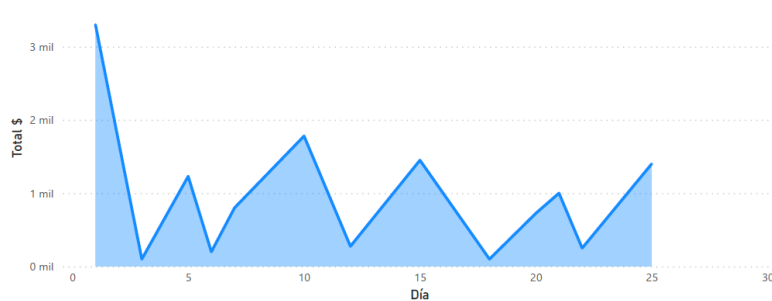
420,33

Ticket Promedio \$

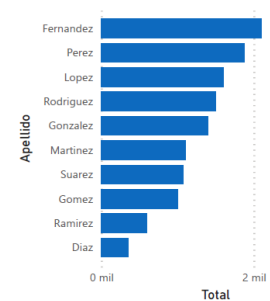
Productos Vendidos por Categoría



Total \$ por Día



Ranking de Clientes



Archivo fuente Power BI

El directorio 'power_bi/' contiene el archivo del proyecto de Power BI y capturas de pantalla del Dashboard.

La visualización del diagrama ER brindado por Power BI se puede consultar en el Anexo 5.

7. Anexos

En esta sección se incluyen los Scripts SQL utilizados para la creación y manipulación de la base de datos, así como capturas de pantalla que muestran las visualizaciones de las tablas, proporcionando una visión detallada del proceso de implementación y análisis.

Anexo 1: Creación del esquema de la base de datos [create_database.sql]

```
-- Crear la base de datos llamada "DeRemate"
CREATE DATABASE DeRemate;

-- Seleccionar la base de datos "DeRemate"
USE DeRemate;

-- Crear tabla Customers para almacenar los datos de los clientes
CREATE TABLE Customers (
    ID INT PRIMARY KEY,
    Nombre NVARCHAR(50),
    Apellido NVARCHAR(50),
    Email NVARCHAR(100),
    FechaRegistro DATE
);

SELECT * FROM Customers;

-- Crear tabla Products para almacenar los datos de los productos
CREATE TABLE Products (
    ID INT PRIMARY KEY,
    Nombre NVARCHAR(100),
    Categoria NVARCHAR(50),
    Precio DECIMAL(10, 2)
);

SELECT * FROM Products;

-- Crear tabla Sales para registrar las ventas, relacionando productos y clientes
CREATE TABLE Sales (
    ID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    Fecha DATE,
    Cantidad INT,
    Total DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(ID),
    FOREIGN KEY (ProductID) REFERENCES Products(ID)
);

SELECT * FROM Sales;
```

Anexo 2: Inserción de datos en la base de datos [populate_data.sql]

```
-- Seleccionar la base de datos "DeRemate"
USE DeRemate;

-- Insertar datos en la tabla Customers
INSERT INTO Customers (ID, Nombre, Apellido, Email, FechaRegistro) VALUES
(1, 'Juan', 'Perez', 'juan.perez@gmail.com', '2023-01-15'),
(2, 'Ana', 'Gomez', 'ana.gomez@gmail.com', '2023-02-20'),
(3, 'Luis', 'Martinez', 'luis.martinez@outlook.com', '2023-03-10'),
(4, 'Maria', 'Lopez', 'maria.lopez@gmail.com', '2023-04-05'),
(5, 'Carlos', 'Fernandez', 'carlos.fernandez@hotmail.com', '2023-05-22'),
(6, 'Sofia', 'Ramirez', 'sofia.ramirez@outlook.com', '2023-06-15'),
(7, 'Miguel', 'Suarez', 'miguel.suarez@fibertel.com', '2023-07-01'),
(8, 'Laura', 'Gonzalez', 'laura.gonzalez@argentina.com', '2023-07-20'),
(9, 'Javier', 'Rodriguez', 'javier.rodriguez@yahoo.com', '2023-08-05'),
(10, 'Lucia', 'Diaz', 'lucia.diaz@facebook.com', '2023-08-15');

SELECT * FROM Customers;

-- Insertar datos en la tabla Products
INSERT INTO Products (ID, Nombre, Categoria, Precio) VALUES
(1, 'Laptop', 'Electronica', 1200.00),
(2, 'Smartphone', 'Electronica', 800.00),
(3, 'Tablet', 'Electronica', 500.00),
(4, 'Monitor', 'Electronica', 300.00),
(5, 'Teclado', 'Accesorios', 50.00),
(6, 'Mouse', 'Accesorios', 30.00),
(7, 'Impresora', 'Electronica', 200.00),
(8, 'Camara', 'Fotografia', 450.00),
(9, 'Auriculares', 'Accesorios', 100.00),
(10, 'Smartwatch', 'Electronica', 250.00),
(11, 'Parlante Bluetooth', 'Accesorios', 150.00),
(12, 'Powerbank', 'Accesorios', 75.00),
(13, 'Televisor', 'Electronica', 1000.00),
(14, 'Microondas', 'Electrodomesticos', 200.00),
(15, 'Heladera', 'Electrodomesticos', 1200.00),
(16, 'Lavarropas', 'Electrodomesticos', 800.00),
(17, 'AirFryer', 'Electrodomesticos', 700.00),
(18, 'Cafetera', 'Electrodomesticos', 150.00),
(19, 'Ventilador', 'Electrodomesticos', 100.00),
(20, 'Licuadora', 'Electrodomesticos', 80.00);

SELECT * FROM Products;

-- Insertar datos en la tabla Sales
INSERT INTO Sales (ID, CustomerID, ProductID, Fecha, Cantidad, Total) VALUES
(1, 1, 1, '2023-06-01', 1, 1200.00),
(2, 1, 5, '2023-06-01', 2, 100.00),
(3, 2, 2, '2023-06-05', 1, 800.00),
(4, 2, 6, '2023-06-05', 1, 30.00),
(5, 3, 3, '2023-06-10', 1, 500.00),
(6, 3, 7, '2023-06-12', 1, 200.00),
(7, 4, 4, '2023-06-15', 2, 600.00),
(8, 4, 9, '2023-06-15', 1, 100.00),
(9, 5, 8, '2023-06-20', 1, 450.00),
(10, 5, 10, '2023-06-22', 1, 250.00),
(11, 1, 3, '2023-07-01', 1, 500.00),
(12, 2, 9, '2023-07-03', 1, 100.00),
(13, 3, 10, '2023-07-05', 1, 250.00),
(14, 4, 2, '2023-07-07', 1, 800.00),
(15, 5, 1, '2023-07-10', 1, 1200.00),
(16, 6, 4, '2023-07-15', 1, 300.00);
```



```
(17, 6, 11, '2023-07-15', 2, 300.00),  
(18, 7, 12, '2023-07-20', 1, 75.00),  
(19, 7, 13, '2023-07-21', 1, 1000.00),  
(20, 8, 14, '2023-07-25', 1, 200.00),  
(21, 8, 15, '2023-07-25', 1, 1200.00),  
(22, 9, 16, '2023-08-01', 1, 800.00),  
(23, 9, 17, '2023-08-01', 1, 700.00),  
(24, 10, 18, '2023-08-05', 1, 150.00),  
(25, 10, 19, '2023-08-06', 2, 200.00),  
(26, 1, 20, '2023-08-10', 1, 80.00),  
(27, 2, 12, '2023-08-12', 1, 75.00),  
(28, 3, 18, '2023-08-15', 1, 150.00),  
(29, 4, 19, '2023-08-18', 1, 100.00),  
(30, 5, 14, '2023-08-20', 1, 200.00);
```

```
SELECT * FROM Sales;
```

Anexo 3: Consultas de análisis [queries_analysis.sql]

```
-- Seleccionar la base de datos "DeRemate"
USE DeRemate;

-- Listar los 5 productos mas vendidos por cantidad total
SELECT TOP 5
    Products.Nombre AS NombreProducto,
    SUM(Sales.Cantidad) AS TotalVendido
FROM
    Sales
JOIN
    Products ON Sales.ProductID = Products.ID
GROUP BY
    Products.Nombre
ORDER BY
    TotalVendido DESC;

-- Encontrar los 5 clientes que han gastado mas en total
SELECT TOP 5
    Customers.Nombre AS Nombre,
    Customers.Apellido AS Apellido,
    SUM(Sales.Total) AS GastoTotal
FROM
    Sales
JOIN
    Customers ON Sales.CustomerID = Customers.ID
GROUP BY
    Customers.Nombre, Customers.Apellido
ORDER BY
    GastoTotal DESC;

-- Obtener el ingreso total por categoria de producto
SELECT
    Products.Categoria AS Categoria,
    SUM(Sales.Total) AS IngresoTotal
FROM
    Sales
JOIN
    Products ON Sales.ProductID = Products.ID
GROUP BY
    Products.Categoria
ORDER BY
    IngresoTotal DESC;

-- Calcular el promedio de ventas diarias (alternativa 1)
SELECT
    AVG(DailySales) AS PromedioVentasDiarias
FROM (
    SELECT
        Fecha,
        SUM(Total) AS DailySales
    FROM
        Sales
    GROUP BY
        Fecha
) AS DailyTotals;

-- Calcular el promedio de ventas diarias (alternativa 2)
SELECT
    SUM(Total) / COUNT(DISTINCT Fecha) AS PromedioVentasDiarias
FROM
    Sales;
```

Anexo 4: Procedimientos almacenados [queries_store_procedures.sql]

```
-- Seleccionar la base de datos "DeRemate"  
USE DeRemate;
```

```
-- Crear un procedimiento almacenado que devuelva todas las compras de un cliente específico
```

```
CREATE PROCEDURE ObtenerComprasPorCliente  
    @CustomerID INT  
AS  
BEGIN  
    SELECT  
        Sales.ID AS IDVenta,  
        Products.Nombre AS NombreProducto,  
        Sales.Fecha,  
        Sales.Cantidad,  
        Sales.Total  
    FROM  
        Sales  
    JOIN  
        Products ON Sales.ProductID = Products.ID  
    WHERE  
        Sales.CustomerID = @CustomerID;  
END;
```

```
-- Ejecutar el procedimiento almacenado para obtener las compras del cliente con ID 1  
EXEC ObtenerComprasPorCliente @CustomerID = 1;
```

```
-- Crear un procedimiento almacenado que devuelva las ventas totales en un rango de fechas
```

```
CREATE PROCEDURE ObtenerVentasPorRangoDeFechas  
    @FechaInicio DATE,  
    @FechaFin DATE  
AS  
BEGIN  
    SELECT  
        SUM(Total) AS VentasTotales  
    FROM  
        Sales  
    WHERE  
        Fecha BETWEEN @FechaInicio AND @FechaFin;  
END;
```

```
-- Ejecutar el procedimiento almacenado para obtener las ventas totales entre 1 de junio y 1 de julio de 2023  
EXEC ObtenerVentasPorRangoDeFechas @FechaInicio = '2023-06-01', @FechaFin = '2023-07-01';
```

Anexo 5: Diagrama ER brindado por Power BI

