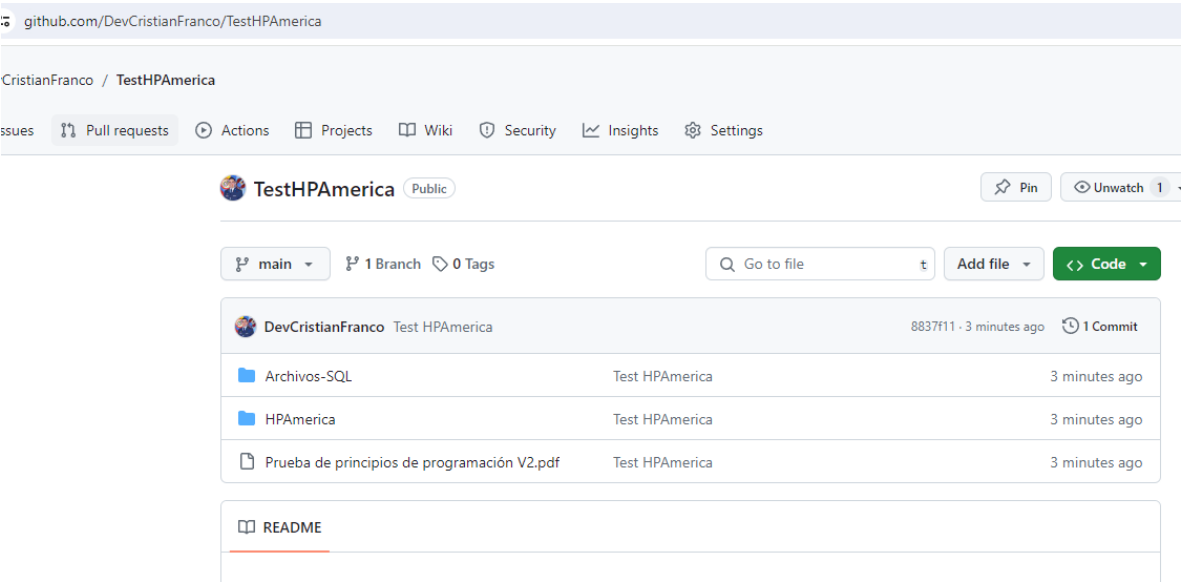


LINK REPO GIT: <https://github.com/DevCristianFranco/TestHPAmerica>



DOCUMENTOS ANEXOS

Archivos-SQL (CREATED STORE PRODUCERS)

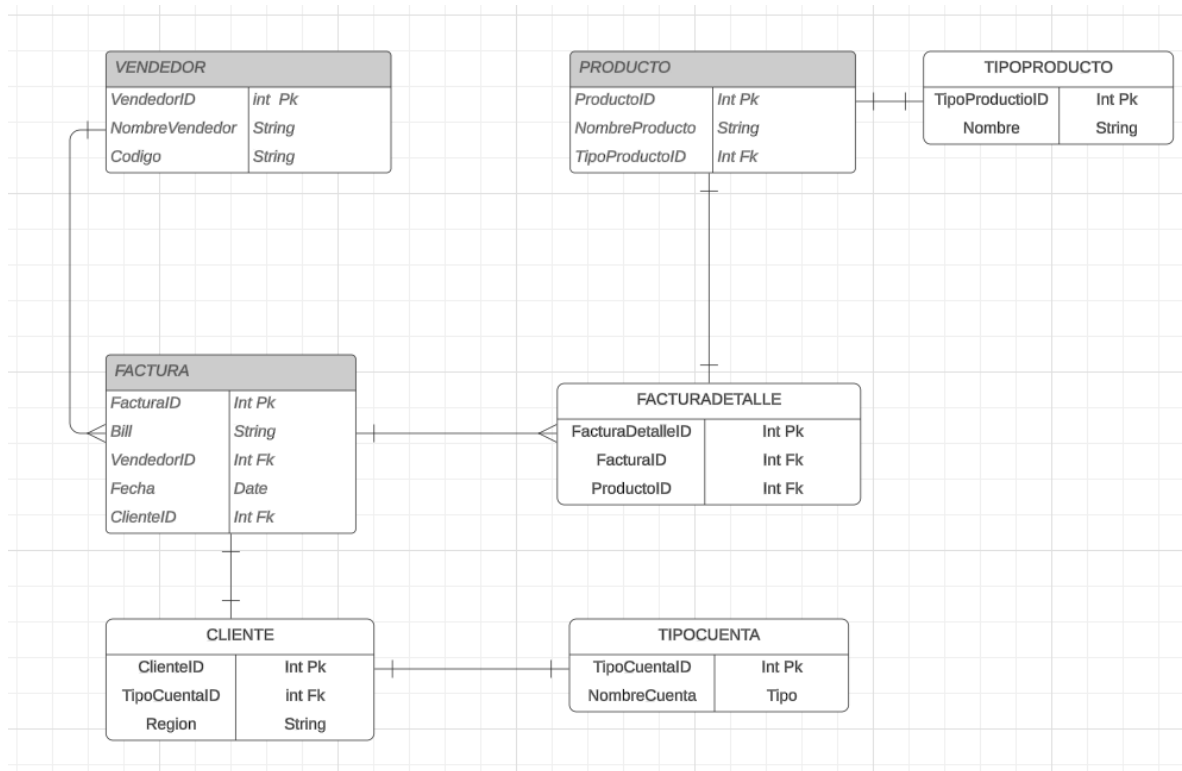
HPAmerica (Solve in Pseudo-Code)

Test-CristianFranco-DEV					Buscar en Test-CristianFranco-DEV	
Nombre		Fecha de modificación	Tipo	Tamaño		
Archivos-SQL		6/25/2024 1:07 PM	Carpeta de archivos			
HPAmerica		6/25/2024 1:07 PM	Carpeta de archivos			

CAPTURAS Y DESARROLLO TEST

BY: Cristian Leonardo Franco -Dev Junior

1.CREATE AN E-R MODEL



2. CREATE STORED PROCEDURES (SP) FOR:

- 2.1) How much Hardware and Software was sold for the global Accounts?

```
CREATE OR ALTER PROCEDURE CalcularVentasGlobales
AS
BEGIN
    -- Crear una tabla temporal para almacenar los resultados
    CREATE TABLE #VentasGlobales (
        Producto NVARCHAR(50),
        CantidadVendida DECIMAL(18, 2)
    );

    -- Insertar las ventas de hardware
    INSERT INTO #VentasGlobales (Producto, CantidadVendida)
    SELECT 'Hardware', SUM(FD.Cantidad)
```

```

FROM FACTURA F
INNER JOIN FACTURADETALLE FD ON F.FacturaID = FD.FacturaID
INNER JOIN PRODUCTO P ON FD.ProductoID = P.ProductoID
INNER JOIN TIPOPRODUCTO TP ON P.TipoProductoID = TP.TipoProductoID
INNER JOIN CLIENTE C ON F.ClienteID = C.ClienteID
INNER JOIN TIPOCUENTA TC ON C.TipoCuentaID = TC.TipoCuentaID
WHERE TP.Nombre = 'Hardware' AND TC.NombreCuenta = 'Global';

-- Insertar las ventas de software
INSERT INTO #VentasGlobales (Producto, CantidadVendida)
SELECT 'Software', SUM(FD.Cantidad)
FROM FACTURA F
INNER JOIN FACTURADETALLE FD ON F.FacturaID = FD.FacturaID
INNER JOIN PRODUCTO P ON FD.ProductoID = P.ProductoID
INNER JOIN TIPOPRODUCTO TP ON P.TipoProductoID = TP.TipoProductoID
INNER JOIN CLIENTE C ON F.ClienteID = C.ClienteID
INNER JOIN TIPOCUENTA TC ON C.TipoCuentaID = TC.TipoCuentaID
WHERE TP.Nombre = 'Software' AND TC.NombreCuenta = 'Global';

-- Seleccionar los resultados finales
SELECT Producto, CantidadVendida
FROM #VentasGlobales;

-- Limpiar la tabla temporal
DROP TABLE #VentasGlobales;
END;
GO

```

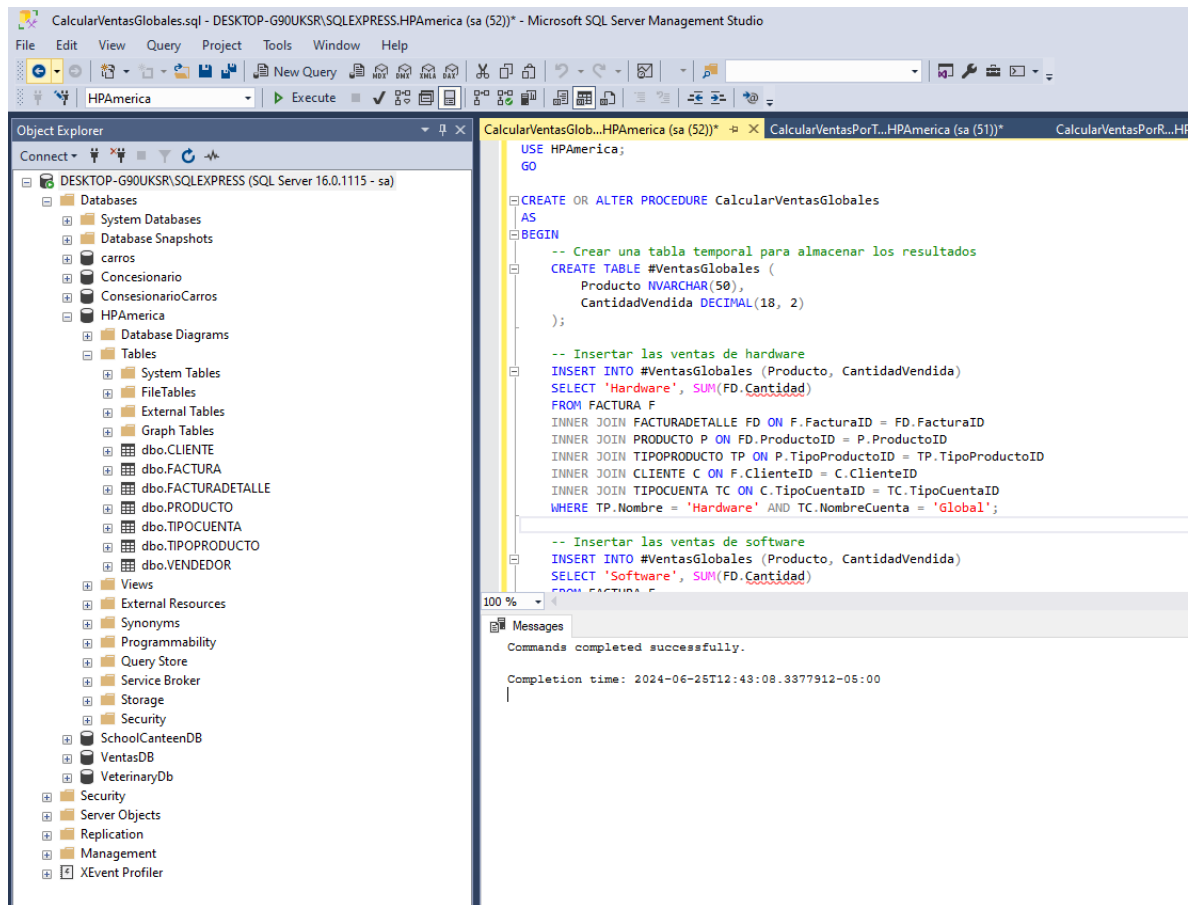


Figure 1 CalcularVentasGlobales.sql

CalcularVentasGlobales.sql - DESKTOP-G90UKSR\SQLEXPRESS.HPAmerica (sa (52))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

HPAmerica Execute

Object Explorer

Connect

DESKTOP-G90UKSR\SQLEXPRESS (SQL Server 16.0.1115 - sa)

- Databases
 - System Databases
 - Database Snapshots
 - carros
 - Concesionario
 - ConcesionarioCarros
 - HPAmerica
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.CLIENTE
 - dbo.FACTURA
 - dbo.FACTURADETALLE
 - dbo.PRODUCTO
 - dbo.TIPOCUENTA
 - dbo.TIPOPUESTO
 - dbo.VENDEDOR
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security
 - SchoolCanteenDB
 - VentasDB
 - VeterinaryDb
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

CalcularVentasGlob...HPAmerica (sa (52))*

EXEC CalcularVentasGlobales;

100 %

Results Messages

	Producto	CantidadVendida
1	Hardware	8.00
2	Software	11.00

Figure 23 Ejecucion y Resultado CalcularVentasGlobales

- 2.2)How much was sold by Region?

```
CREATE PROCEDURE CalcularVentasPorRegion
AS
BEGIN
    -- Crear una tabla temporal para almacenar los resultados
    CREATE TABLE #VentasPorRegion (
        Region NVARCHAR(50),
        Producto NVARCHAR(50),
        CantidadVendida DECIMAL(18, 2)
    );

    -- Insertar las ventas por región
    INSERT INTO #VentasPorRegion (Region, Producto, CantidadVendida)
    SELECT C.Region AS Region, TP.Nombre AS Producto, SUM(FD.Cantidad) AS
    CantidadVendida
    FROM FACTURA F
    INNER JOIN FACTURADETALLE FD ON F.FacturaID = FD.FacturaID
    JOIN PRODUCTO P ON FD.ProductoID = P.ProductoID
    JOIN TIPOPRODUCTO TP ON P.TipoProductoID = TP.TipoProductoID
    INNER JOIN CLIENTE C ON F.ClienteID = C.ClienteID
    GROUP BY C.Region, TP.Nombre;

    -- Seleccionar los resultados finales
    SELECT Region, Producto, CantidadVendida
    FROM #VentasPorRegion;

    -- Limpiar la tabla temporal
    DROP TABLE #VentasPorRegion;
END;
```

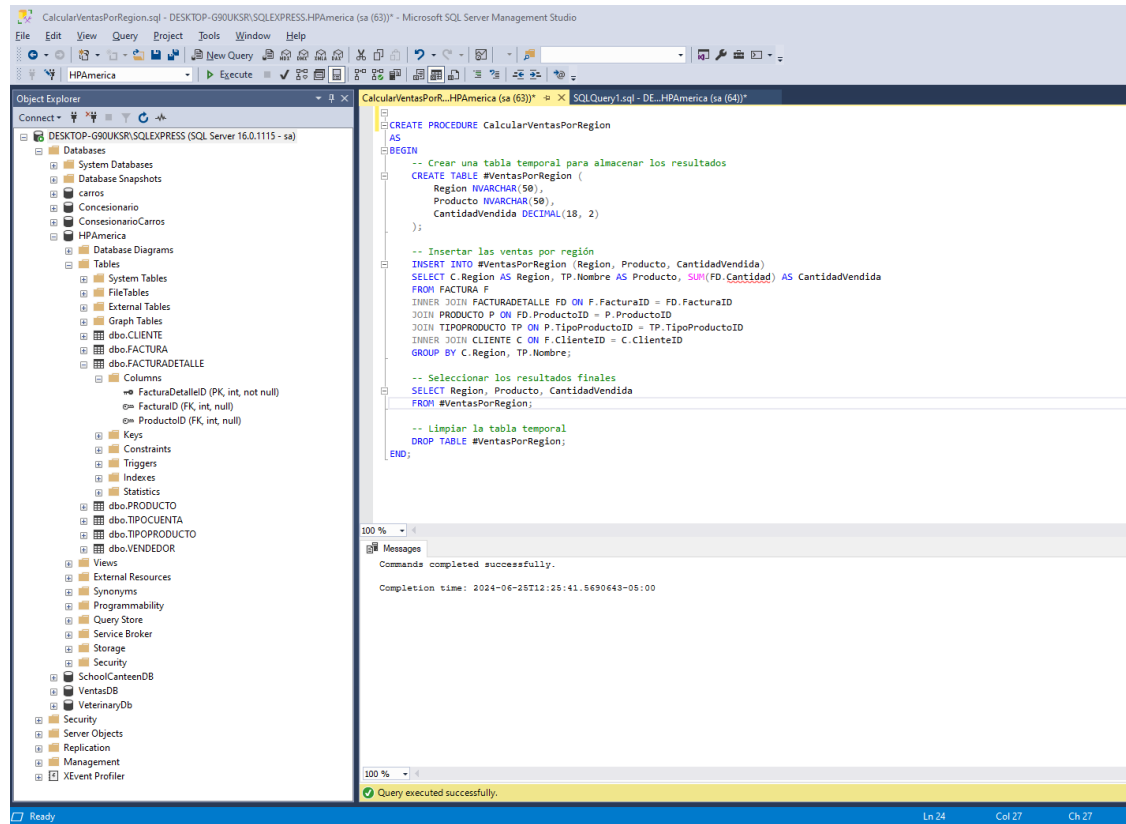


Figure 3 CalcularVentasPorRegion.sql

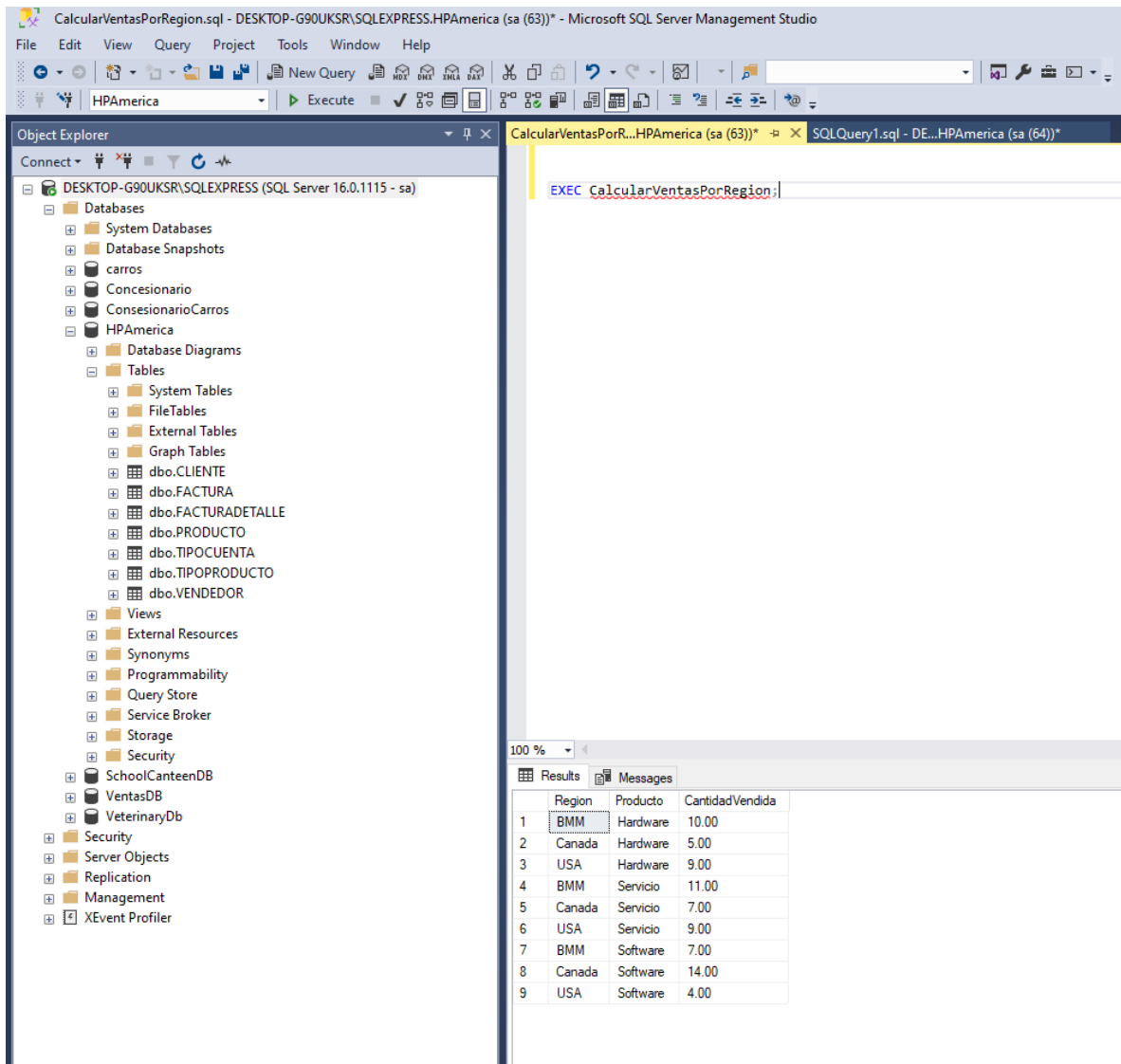


Figure 4 Ejecucion y Resultado VentasPorRegion.sql

- 2.3)How much was sold by kind of Product?

```
CREATE PROCEDURE CalcularVentasPorTipoProducto
AS
BEGIN
    -- Crear una tabla temporal para almacenar los resultados
    CREATE TABLE #VentasPorTipoProducto (
        TipoProducto NVARCHAR(50),
        CantidadVendida DECIMAL(18, 2)
    );

    -- Insertar las ventas por tipo de producto
    INSERT INTO #VentasPorTipoProducto (TipoProducto, CantidadVendida)
    SELECT TP.Nombre AS TipoProducto, SUM(FD.Cantidad) AS CantidadVendida
    FROM FACTURA F
    INNER JOIN FACTURADETALLE FD ON F.FacturaID = FD.FacturaID
    JOIN PRODUCTO P ON FD.ProductoID = P.ProductoID
    JOIN TIPOPRODUCTO TP ON P.TipoProductoID = TP.TipoProductoID
    GROUP BY TP.Nombre;

    -- Seleccionar los resultados finales
    SELECT TipoProducto, CantidadVendida
    FROM #VentasPorTipoProducto;

    -- Limpiar la tabla temporal
    DROP TABLE #VentasPorTipoProducto;
END;
```

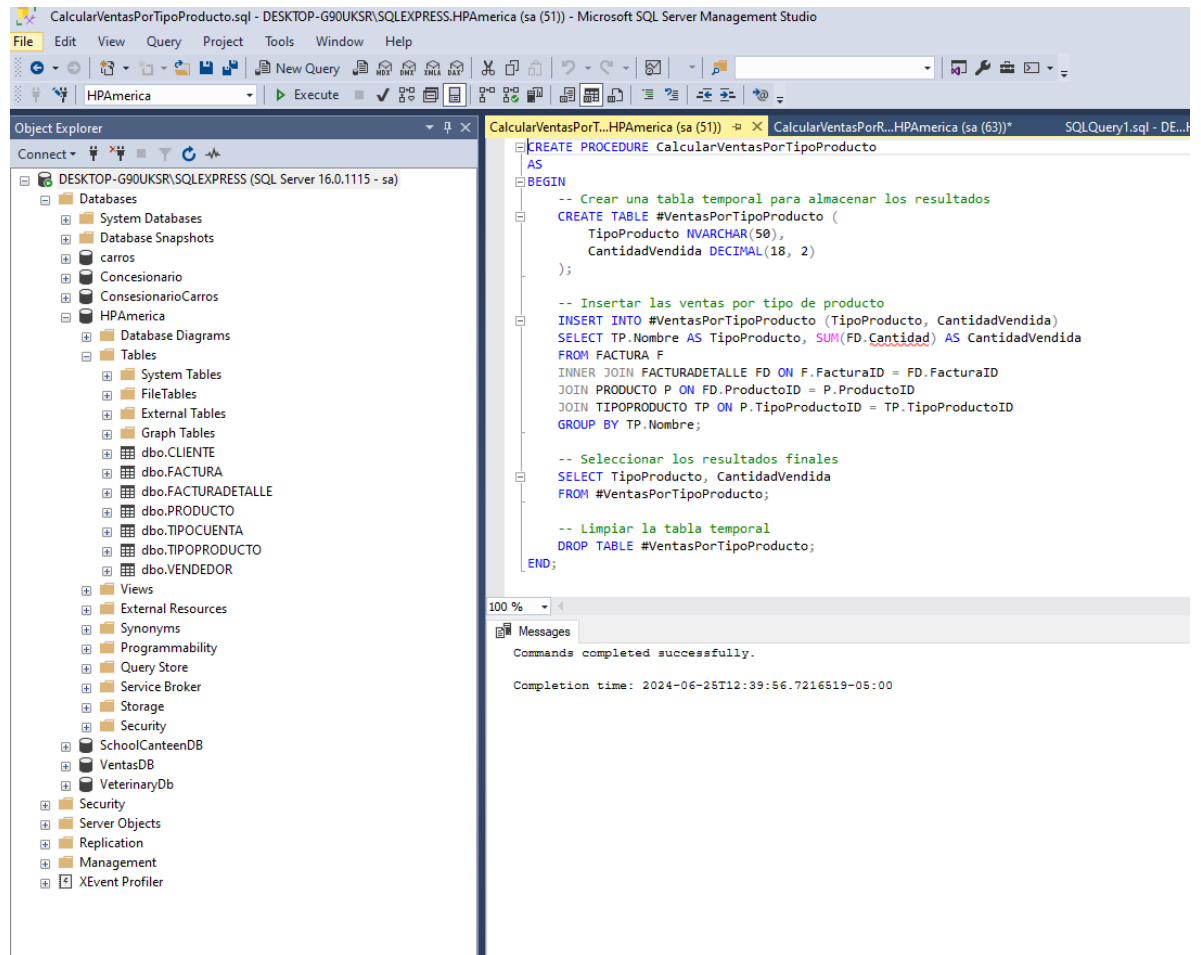


Figure 5CalcularVentasPorTipoProducto.sql

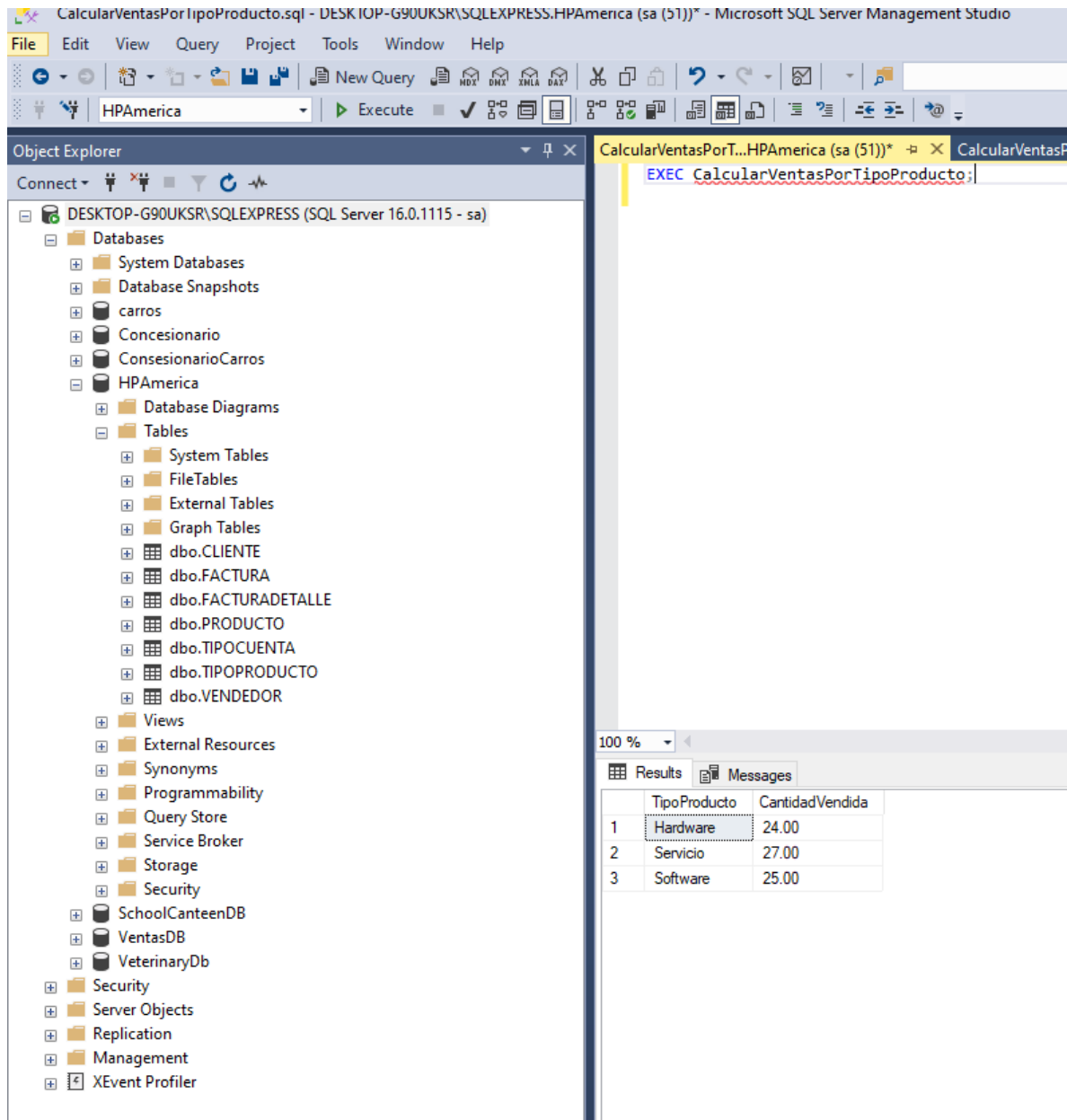


Figure 6 Ejecucion y ResultadoCalcularVentasPorTipoProducto

3. Solve in Pseudo-Code the following scenarios

3.1) Suppose that the sales numbers are in a flat file formatted as View 1, create a process to read that data.

<< HPAmerica > bin > Debug > net8.0

Buscar en net8.0







Nombre	Fecha de modificación	Tipo	Tamaño
 HPAmerica.deps.json	6/25/2024 12:09 PM	JSON File	1 KB
 HPAmerica.dll	6/25/2024 12:20 PM	Extensión de la ap...	8 KB
 HPAmerica.exe	6/25/2024 12:20 PM	Aplicación	140 KB
 HPAmerica.pdb	6/25/2024 12:20 PM	Program Debug D...	12 KB
 HPAmerica.runtimeconfig.json	6/25/2024 12:09 PM	JSON File	1 KB
 ventas.csv	6/25/2024 12:19 PM	Archivo de valores...	1 KB

Figure 7 Creacion Archivo CSV (ventas.csv)

Datos ventas csv

FacturalID,Producto,Cantidad,Fecha,ClienteID,Region
1,Laptop,5,2024-06-01,1,USA
2,Desktop,3,2024-06-02,2,Canada
3,Mouse,10,2024-06-03,3,BMM
4,Keyboard,4,2024-06-04,4,USA
5,Monitor,2,2024-06-05,5,Canada
6,Office Suite,7,2024-06-06,6,BMM
7,Antivirus,1,2024-06-07,7,USA
8,Operating System,6,2024-06-08,8,Canada
9,Design Software,3,2024-06-09,1,BMM
10,Accounting Software,8,2024-06-10,2,USA
11,Consultoría,2,2024-06-11,3,Canada
12,Soporte Técnico,5,2024-06-12,4,BMM
13,Mantenimiento,7,2024-06-13,5,USA
14,Instalación,9,2024-06-14,6,Canada
15,Capacitación,4,2024-06-15,7,BMM

3.2) Obtain the sales average for a specific time interval.

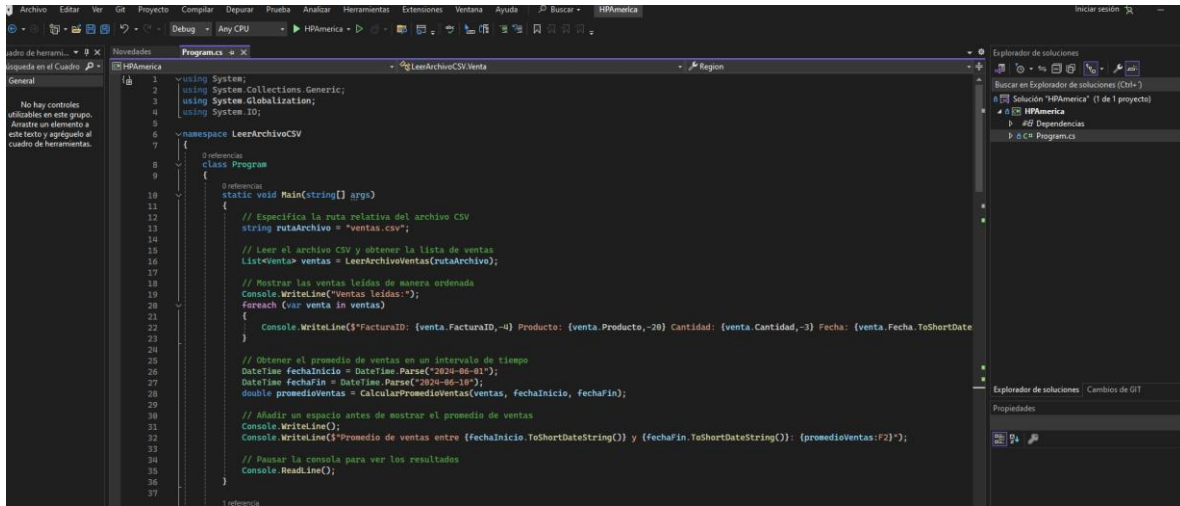


Figure 8 Proyecto consola C# HPAmerica

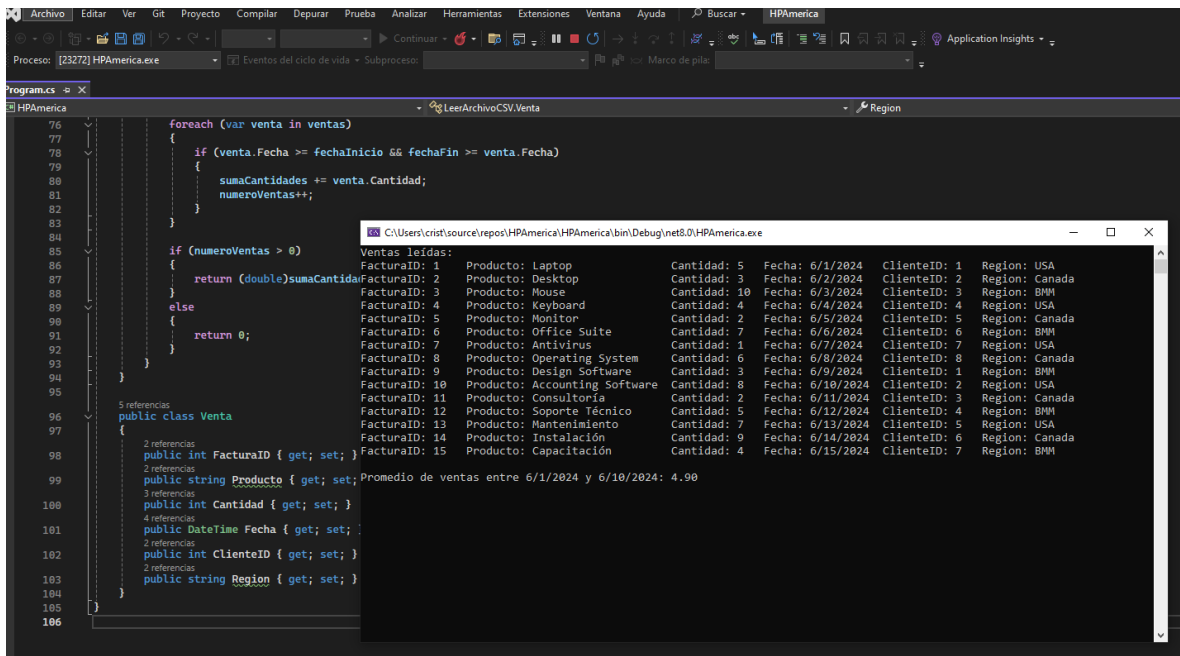


Figure 9 Ejecucion y Resultado

CODIGO:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;

namespace LeerArchivoCSV
{
    class Program
    {
        static void Main(string[] args)
        {
            // Especifica la ruta relativa del archivo CSV
            string rutaArchivo = "ventas.csv";

            // Leer el archivo CSV y obtener la lista de ventas
            List<Venta> ventas = LeerArchivoVentas(rutaArchivo);

            // Mostrar las ventas leídas de manera ordenada
            Console.WriteLine("Ventas leídas:");
            foreach (var venta in ventas)
            {
                Console.WriteLine($"FacturaID: {venta.FacturaID,-4} Producto:
{venta.Producto,-20} Cantidad: {venta.Cantidad,-3} Fecha:
{venta.Fecha.ToShortDateString(),-10} ClienteID: {venta.ClienteID,-3} Region:
{venta.Region}");
            }

            // Obtener el promedio de ventas en un intervalo de tiempo
            DateTime fechaInicio = DateTime.Parse("2024-06-01");
            DateTime fechaFin = DateTime.Parse("2024-06-10");
            double promedioVentas = CalcularPromedioVentas(ventas, fechaInicio,
fechaFin);

            // Añadir un espacio antes de mostrar el promedio de ventas
            Console.WriteLine();
            Console.WriteLine($"Promedio de ventas entre
{fechaInicio.ToShortDateString()} y {fechaFin.ToShortDateString()}:
{promedioVentas:F2}");

            // Pausar la consola para ver los resultados
            Console.ReadLine();
        }

        static List<Venta> LeerArchivoVentas(string rutaArchivo)
        {
            var ventas = new List<Venta>();
            try
            {
                using (var reader = new StreamReader(rutaArchivo))
                {
                    string linea;
                    reader.ReadLine(); // Leer y descartar la cabecera

                    while ((linea = reader.ReadLine()) != null)
                    {
                        var valores = linea.Split(',');
                    }
                }
            }
            catch { }
        }
    }
}
```

```

        var venta = new Venta
        {
            FacturaID = int.Parse(valores[0]),
            Producto = valores[1],
            Cantidad = int.Parse(valores[2]),
            Fecha = DateTime.ParseExact(valores[3], "yyyy-MM-
dd", CultureInfo.InvariantCulture),
            ClienteID = int.Parse(valores[4]),
            Region = valores[5]
        };
        ventas.Add(venta);
    }
}
}
catch (Exception ex)
{
    Console.WriteLine($"Error al leer el archivo: {ex.Message}");
}
return ventas;
}

static double CalcularPromedioVentas(List<Venta> ventas, DateTime
fechaInicio, DateTime fechaFin)
{
    int sumaCantidades = 0;
    int numeroVentas = 0;

    foreach (var venta in ventas)
    {
        if (venta.Fecha >= fechaInicio && fechaFin >= venta.Fecha)
        {
            sumaCantidades += venta.Cantidad;
            numeroVentas++;
        }
    }

    if (numeroVentas > 0)
    {
        return (double)sumaCantidades / numeroVentas;
    }
    else
    {
        return 0;
    }
}

}

public class Venta
{
    public int FacturaID { get; set; }
    public string Producto { get; set; }
    public int Cantidad { get; set; }
    public DateTime Fecha { get; set; }
    public int ClienteID { get; set; }
    public string Region { get; set; }
}
}

```