

**TECNOLÓGICO DE ESTUDIOS SUPERIORES DE  
CHIMALHUACÁN**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**



# **SEGUNDA IMPLEMENTACION BACKEND DE LA BASE DE DATOS RETAIL CON T-SQL**

**•JIMÉNEZ COLORADO MELANIE FERNANDA**

**•TOLENTINO BAUTISTA JOSÉ ANDRÉS**

**•NOGUEZ CHAVARRIA PABLO YOEL**

**PROFESOR: Dr. Jiménez Alfaro Abraham Jorge**

**ASIGNATURA: Minería de Datos**

**GRUPO: “7ISC23”**

**2025-2**

# INTRODUCCION

Este proyecto tiene como propósito reforzar el manejo de T-SQL dentro del ámbito de la minería de datos, a través de la segunda etapa de desarrollo del backend para una base de datos de tipo Retail. Se pretende perfeccionar consultas, procedimientos almacenados y esquemas que faciliten la obtención de conocimiento significativo a partir de grandes conjuntos de datos transaccionales. El enfoque se dirige a incrementar la eficiencia, la capacidad de escalamiento y el poder analítico del sistema, incorporando prácticas adecuadas de diseño y programación en SQL Server.

Asimismo, busca afianzar el dominio práctico en el manejo de bases de datos relacionales mediante una implementación avanzada del backend para un entorno Retail. Empleando T-SQL como herramienta central, se procura optimizar el desempeño de consultas, procedimientos almacenados y estructuras de datos que permitan extraer información relevante para apoyar la toma de decisiones. Esta segunda fase se orienta a potenciar la eficiencia, la escalabilidad y las capacidades analíticas del sistema, integrando técnicas de minería de datos para identificar patrones ocultos dentro de los registros transaccionales.

## DESARROLLO

Durante la elaboración del proyecto se registró de manera detallada la organización y el comportamiento interno de la base de datos Retail. En primer lugar, se creó el diagrama entidad–relación (ER), en el cual se definieron las entidades principales del sistema —Clientes, Productos, Ventas, Empleados y Categorías— junto con sus interacciones. Este diagrama facilitó la comprensión de cómo se vinculan los datos y funcionó como fundamento para la construcción de las tablas en SQL Server.

Database snapshots

- AdventureWorks2017
- DW\_Planta
- PlantAndHealth
- PRUEBA
- Venta
- VentasDM
- pluginfile
- Database Diagrams
- Tables
  - System Tables
  - FileTables
  - External Tables
  - Graph Tables
- dbo.Categories
- dbo.CustomerCustomerDi
- dbo.CustomerDemograp
- dbo.Customers
- dbo.Employees
- dbo.EmployeeTerritories
- dbo.Order Details
- dbo.Orders
- dbo.Products
- dbo.Region
- dbo.Shippers
- dbo.Suppliers
- dbo.Territories
- Views
- External Resources
- Synonyms
- Programmability
- Antora Rembar

```
-- Ver las relaciones entre tablas
SELECT
    fk.name AS 'Foreign Key',
    tp.name AS 'Parent Table',
    cp.name AS 'Parent Column',
    tr.name AS 'Referenced Table',
    cr.name AS 'Referenced Column'
FROM sys.foreign_keys fk
INNER JOIN sys.tables tp ON fk.parent_object_id = tp.object_id
INNER JOIN sys.tables tr ON fk.referenced_object_id = tr.object_id
INNER JOIN sys.foreign_key_columns fkc ON fkc.constraint_object_id = fk.object_id
INNER JOIN sys.columns cp ON fkc.parent_column_id = cp.column_id AND fkc.parent_object_id = cp.object_id
INNER JOIN sys.columns cr ON fkc.referenced_column_id = cr.column_id AND fkc.referenced_object_id = cr.object_id;
```

100 %

Results Messages

Column Name	Data Type	Is Nullable
CustomerID	nchar	NO
CompanyName	nvarchar	NO
ContactName	nvarchar	YES
ContactTitle	nvarchar	YES
Address	nvarchar	YES
City	nvarchar	YES
Region	nvarchar	YES
PostalCode	nvarchar	YES

TABLE NAME

1	syndiagrams
2	Employees
3	Categories
4	Customers
5	Shippers
6	Suppliers
7	Orders

Se desarrolló el script en T-SQL para la creación de las tablas, incorporando la definición de claves primarias y foráneas, los tipos de datos y las restricciones de integridad. De esta forma, el modelo conceptual se convirtió en estructuras físicas dentro del motor de la base de datos.

Connect

JAEL (SQL Server 14.0.1000.169 - Micr

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks2017
  - DW\_Planta
  - PlantAndHealth
  - PRUEBA
  - Venta
  - VentasDM
  - pluginfile
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
  - dbo.Categories
  - dbo.CustomerCustomerDi
  - dbo.CustomerDemograp
  - dbo.Customers
  - dbo.Employees
  - dbo.EmployeeTerritories
  - dbo.Order Details
  - dbo.Orders

```
-- Insertar datos en Categories (si no existen)
IF NOT EXISTS (SELECT 1 FROM Categories WHERE CategoryID = 1)
BEGIN
    INSERT INTO Categories (CategoryID, CategoryName, Description) VALUES
    (1, 'Beverages', 'Soft drinks, coffees, teas, beers, and ales'),
    (2, 'Condiments', 'Sweet and savory sauces, relishes, spreads, and seasonings'),
    (3, 'Confections', 'Desserts, candies, and sweet breads'),
    (4, 'Dairy Products', 'Cheeses'),
    (5, 'Grains/Cereals', 'Breads, crackers, pasta, and cereal'),
    (6, 'Meat/Poultry', 'Prepared meats'),
    (7, 'Produce', 'Dried fruit and bean curd'),
    (8, 'Seafood', 'Seaweed and fish');
END

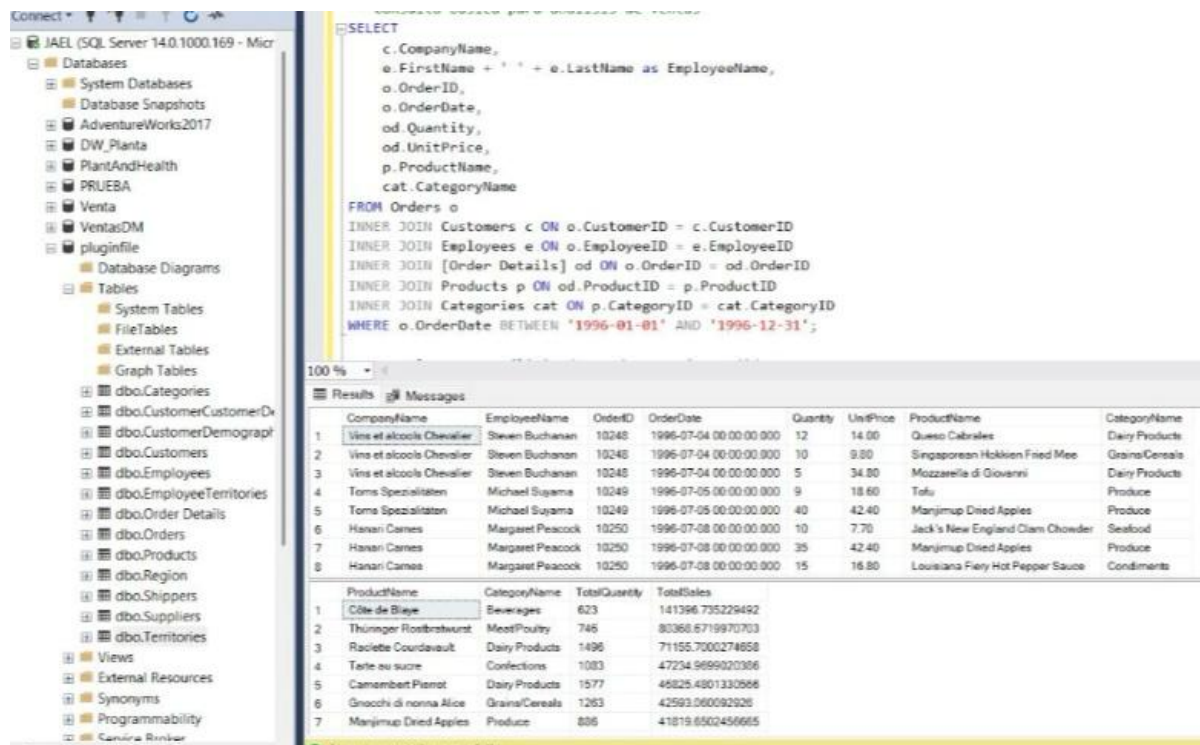
-- Insertar datos en Customers (si no existen)
IF NOT EXISTS (SELECT 1 FROM Customers WHERE CustomerID = 'ALFKI')
BEGIN
    INSERT INTO Customers (CustomerID, CompanyName, ContactName, City, Country) VALUES
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-11-14T00:26:11.7110276-06:00



```

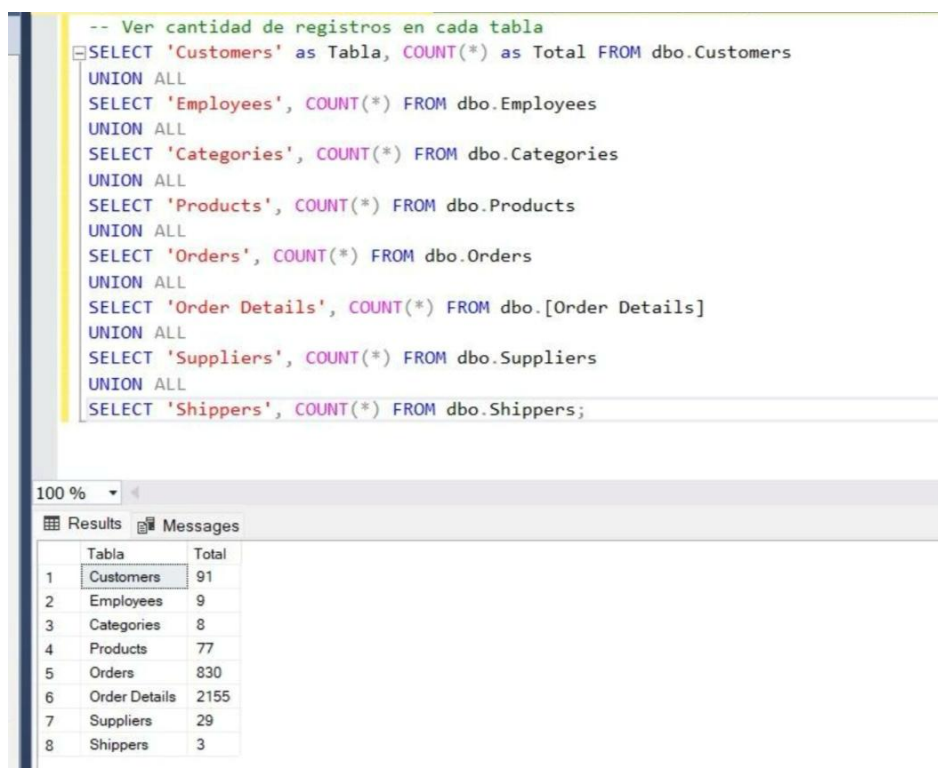
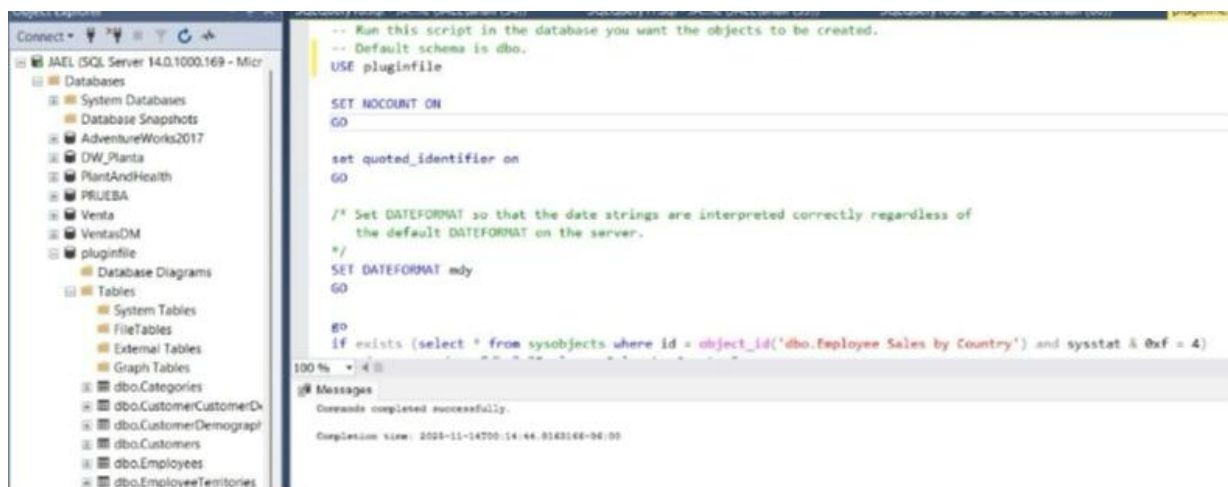
SELECT
    c.CompanyName,
    e.FirstName + ' ' + e.LastName as EmployeeName,
    o.OrderID,
    o.OrderDate,
    od.Quantity,
    od.UnitPrice,
    p.ProductName,
    cat.CategoryName
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
INNER JOIN Employees e ON o.EmployeeID = e.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Products p ON od.ProductID = p.ProductID
INNER JOIN Categories cat ON p.CategoryID = cat.CategoryID
WHERE o.OrderDate BETWEEN '1996-01-01' AND '1996-12-31';

```

	CompanyName	EmployeeName	OrderID	OrderDate	Quantity	UnitPrice	ProductName	CategoryName
1	Vins et alcools Chevalier	Steven Buchanan	10248	1996-07-04 00:00:00.000	12	14.00	Queso Cabrales	Dairy Products
2	Vins et alcools Chevalier	Steven Buchanan	10248	1996-07-04 00:00:00.000	10	9.80	Singaporean Hokkien Fried Mee	Grains/Cereals
3	Vins et alcools Chevalier	Steven Buchanan	10248	1996-07-04 00:00:00.000	5	34.80	Mozzarella di Giovanni	Dairy Products
4	Toms Spezialitäten	Michael Suyama	10249	1996-07-05 00:00:00.000	9	18.60	Tofu	Produce
5	Toms Spezialitäten	Michael Suyama	10249	1996-07-05 00:00:00.000	40	42.40	Manjimup Dried Apples	Produce
6	Hanari Carnes	Margaret Peacock	10250	1996-07-08 00:00:00.000	10	7.70	Jack's New England Clam Chowder	Seafood
7	Hanari Carnes	Margaret Peacock	10250	1996-07-08 00:00:00.000	35	42.40	Manjimup Dried Apples	Produce
8	Hanari Carnes	Margaret Peacock	10250	1996-07-08 00:00:00.000	15	16.80	Louisiana Fiery Hot Pepper Sauce	Condiments

	ProductName	CategoryName	TotalQuantity	TotalSales
1	Côte de Blaye	Beverages	623	141396.735229492
2	Thüringer Rostbratwurst	Meat/Poultry	746	80368.6719970703
3	Raclette Courdaival	Dairy Products	1496	71155.7000274658
4	Tarte au sucre	Confections	1083	47234.969920386
5	Camembert Pierrot	Dairy Products	1577	46825.4801330566
6	Gnocchi di nonna Alice	Grains/Cereals	1263	42593.060092926
7	Manjimup Dried Apples	Produce	806	41819.6502456665

Además, se desarrollaron consultas de análisis que permiten aplicar técnicas elementales de minería de datos. Mediante el uso de funciones como SUM, AVG y COUNT, junto con cláusulas como GROUP BY, HAVING y ORDER BY, fue posible detectar patrones relevantes, tales como los productos más solicitados, los clientes con mayor número de compras y las tendencias de ventas según la temporada.



Los resultados obtenidos se organizaron en tablas que facilitan su interpretación, mostrando información como los diez productos más vendidos, los clientes con mayor gasto acumulado y las ventas agrupadas por categoría y mes.

```

        (SELECT SUM(od2.Quantity * od2.UnitPrice * (1 - od2.Discount)))
        FROM dbo.[Order Details] od2) * 100, 2) as PorcentajeTotal
FROM dbo.Categories c
INNER JOIN dbo.Products p ON c.CategoryID = p.CategoryID
INNER JOIN dbo.[Order Details] od ON p.ProductID = od.ProductID
INNER JOIN dbo.Orders o ON od.OrderID = o.OrderID
GROUP BY c.CategoryID, c.CategoryName
ORDER BY VentasTotales DESC;

-- 2. Comportamiento de clientes (Top 10)
SELECT TOP 10
    c.CompanyName as Cliente,
    c.Country as Pais,
    COUNT(DISTINCT o.OrderID) as TotalOrdenes,
    SUM(od.Quantity * od.UnitPrice * (1 - od.Discount)) as TotalGastado,
    AVG(od.Quantity * od.UnitPrice * (1 - od.Discount)) as PromedioPorOrden
FROM dbo.Customers c
INNER JOIN dbo.Orders o ON c.CustomerID = o.CustomerID

```

100 %

Results Messages

Categoria	TotalOrdenes	TotalProductosVendidos	VentasTotales	PorcentajeTotal
Beverages	354	9532	267868.180522919	21.16
Dairy Products	303	9149	234507.285217285	18.53
Confections	295	7906	167357.224831581	13.22
Meat/Poultry	161	4199	163022.359088898	12.88
Seafood	291	7681	131261.73742485	10.37
Condiments	193	5298	106047.084989548	8.38
Produce	129	2990	99984.5800685883	7.9
Grains/Cereals	182	4562	95744.587474823	7.56

Cliente	Pais	TotalOrdenes	TotalGastado	PromedioPorOrden
QUICK-Stop	Germany	28	110277.304977417	1282.29424392345
Ernst Handel	Austria	30	104874.978713989	1028.18606582342
Save-a-lot Markets	USA	31	104361.949920654	899.671982074606
Rattlesnake Canyon Grocery	USA	18	51097.8003330231	719.687328634128
Hungry Owl All-Night Grocers	Ireland	19	49979.9050006866	908.72554546703
Hanari Carnes	Brazil	14	32841.3699417114	1026.29281067848
Königlich Essen	Germany	14	30908.3839836121	792.522666246463

```

-- 4. Productos más vendidos y rentables
SELECT TOP 15
    p.ProductName as Producto,
    c.CategoryName as Categoria,
    SUM(od.Quantity) as TotalVendido,
    ROUND(AVG(od.UnitPrice), 2) as PrecioPromedio,
    SUM(od.Quantity * od.UnitPrice * (1 - od.Discount)) as IngresosTotales,
    p.UnitsInStock as InventarioActual,
    p.UnitsOnOrder as EnPedido
FROM dbo.Products p
INNER JOIN dbo.Categories c ON p.CategoryID = c.CategoryID
INNER JOIN dbo.[Order Details] od ON p.ProductID = od.ProductID
GROUP BY p.ProductID, p.ProductName, c.CategoryName, p.UnitsInStock, p.UnitsOnOrder
ORDER BY IngresosTotales DESC;

-- 5. Análisis de productos discontinuados
SELECT
    c.CategoryName as Categoria,

```

100 %

Results Messages

Producto	Categoria	TotalVendido	PrecioPromedio	IngresosTotales	InventarioActual	EnPedido
Côte de Blaye	Beverages	623	245.93	141396.735229492	17	0
Thüringer Rostbratwurst	Meat/Poultry	746	116.04	80368.6719970703	0	0
Raclette Courdavault	Dairy Products	1496	51.13	71155.7000274658	79	0
Tarte au sucre	Confections	1083	46.41	47234.9699020386	17	0
Camembert Pierrot	Dairy Products	1577	32.13	46825.4801330566	19	0
Gnocchi di nonna Alice	Grains/Cereals	1263	35.42	42593.060092926	21	10
Manjimup Dried Apples	Produce	886	50.55	41819.6502456665	20	0
Alice Mutton	Meat/Poultry	978	36.47	32698.3799819946	0	0

Categoria	TotalProductos	ProductosDescontinuados	PorcentajeDescontinuados
Meat/Poultry	6	4	66.6700000000000
Produce	5	1	20.0000000000000
Grains/Cereals	7	1	14.2900000000000
Beverages	12	1	8.33000000000000
Condiments	12	1	8.33000000000000
Confections	13	0	0.00000000000000
Dairy Products	10	0	0.00000000000000

Query executed successfully.



```
-- 6. Ventas mensuales (tendencias temporales)
```

```
SELECT
```

```
    YEAR(o.OrderDate) as Año,
```

```
    MONTH(o.OrderDate) as Mes,
```

```
    DATENAME(MONTH, o.OrderDate) as NombreMes,
```

```
    COUNT(DISTINCT o.OrderID) as TotalOrdenes,
```

```
    SUM(od.Quantity) as TotalProductos,
```

```
    ROUND(SUM(od.Quantity * od.UnitPrice * (1 - od.Discount)), 2) as VentasTotales
```

```
FROM dbo.Orders o
```

```
INNER JOIN dbo.[Order Details] od ON o.OrderID = od.OrderID
```

```
WHERE o.OrderDate IS NOT NULL
```

```
GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate), DATENAME(MONTH, o.OrderDate)
```

```
ORDER BY Año, Mes;
```

```
-- 7. Cumplimiento de fechas de entrega
```

```
SELECT
```

```
    COUNT(*) as TotalOrdenes,
```

```
    SUM(CASE WHEN o.ShippedDate <= o.RequiredDate THEN 1 ELSE 0 END) as EntregasATiempo,
```

```
    SUM(CASE WHEN o.ShippedDate > o.RequiredDate THEN 1 ELSE 0 END) as EntregasTardias,
```

```
    SUM(CASE WHEN o.ShippedDate > o.RequiredDate THEN 1 ELSE 0 END) as PendientesEnvio,
```

```
    ROUND(SUM(CASE WHEN o.ShippedDate <= o.RequiredDate THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as PorcentajeCumplimiento
```

```
FROM dbo.Orders o
```

```
WHERE o.ShippedDate IS NOT NULL
```

```
GROUP BY
```

```
ORDER BY
```

Año	Mes	NombreMes	TotalOrdenes	TotalProductos	VentasTotales
1996	7	July	22	1462	27861.89
1996	8	August	25	1322	25485.27
1996	9	September	23	1124	26381.4
1996	10	October	26	1738	37515.73
1996	11	November	25	1735	45600.04
1996	12	December	31	2200	45239.63
1997	1	January	33	2401	61258.07
1997	2	February	29	2132	38483.63
1997	3	March	30	1770	38547.22
1997	4	April	31	1912	53032.95
1997	5	May	32	2164	53781.29

TotalOrdenes	EntregasATiempo	EntregasTardias	PendientesEnvio	PorcentajeCumplimiento
830	772	37	21	93.010000000000

```
-- 8. Análisis de datos faltantes
```

```
SELECT
```

```
    'Customers - Region' as Campo,
```

```
    SUM(CASE WHEN Region IS NULL THEN 1 ELSE 0 END) as Nulos,
```

```
    COUNT(*) as Total,
```

```
    ROUND(SUM(CASE WHEN Region IS NULL THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as PorcentajeNulos
```

```
FROM dbo.Customers
```

```
UNION ALL
```

```
SELECT 'Customers - Fax',
```

```
    SUM(CASE WHEN Fax IS NULL THEN 1 ELSE 0 END),
```

```
    COUNT(*),
```

```
    ROUND(SUM(CASE WHEN Fax IS NULL THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2)
```

```
FROM dbo.Customers
```

```
UNION ALL
```

```
SELECT 'Employees - Region',
```

```
    SUM(CASE WHEN Region IS NULL THEN 1 ELSE 0 END),
```

```
    COUNT(*),
```

```
    ROUND(SUM(CASE WHEN Region IS NULL THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2)
```

```
FROM dbo.Employees
```

Campo	Nulos	Total	PorcentajeNulos
Customers - Region	60	91	65.930000000000
Customers - Fax	22	91	24.180000000000
Employees - Region	4	9	44.400000000000

Pais	TotalClientes	Porcentaje
USA	13	14.290000000000
France	11	12.090000000000
Germany	11	12.090000000000
Brazil	9	9.890000000000
UK	7	7.690000000000
Spain	5	5.490000000000
Mexico	5	5.490000000000
Venezu...	4	4.400000000000
Italy	3	3.300000000000
Canada	3	3.300000000000

-- 10. Resumen ejecutivo para el proyecto

```
SELECT
    (SELECT COUNT(*) FROM dbo.Customers) as TotalClientes,
    (SELECT COUNT(*) FROM dbo.Products) as TotalProductos,
    (SELECT COUNT(*) FROM dbo.Orders) as TotalOrdenes,
    (SELECT COUNT(*) FROM dbo.Employees) as TotalEmpleados,
    (SELECT ROUND(SUM(Quantity * UnitPrice * (1 - Discount)), 2)
     FROM dbo.[Order Details]) as VentasTotales,
    (SELECT COUNT(DISTINCT Country) FROM dbo.Customers) as PaísesConClientes,
    (SELECT COUNT(DISTINCT CategoryID) FROM dbo.Products) as CategoríasActivas;
```

00 %

Results Messages

	TotalClientes	TotalProductos	TotalOrdenes	TotalEmpleados	VentasTotales	PaísesConClientes	CategoríasActivas
1	91	77	830	9	1265793.04	21	8



# CONCLUSION

El desarrollo del proyecto permitió entender de forma completa tanto la organización interna como el funcionamiento operativo de la base de datos Retail. Mediante el diseño del modelo entidad-relación, la construcción de tablas en SQL Server, la implementación de procedimientos almacenados y la creación de consultas analíticas, se logró establecer un sistema funcional, coherente y orientado al apoyo en la toma de decisiones.

El estudio de la información generada por la base de datos proporcionó una perspectiva clara sobre los hábitos de consumo, el desempeño de los productos y las tendencias de venta, lo cual evidencia la relevancia de combinar un modelo de datos bien construido con técnicas de análisis apropiadas. En conjunto, este proyecto confirma el valor de contar con bases de datos correctamente estructuradas para respaldar procesos empresariales, mejorar operaciones y facilitar la interpretación estratégica de los resultados.

# BIBLIOGRAFIAS

Coronel, C., & Morris, S. (2017). Database Systems: Design, Implementation, and Management. Cengage Learning.

Elmasri, R., & Navathe, S. (2015). Fundamentals of Database Systems. Pearson.

Date, C. J. (2004). An Introduction to Database Systems. AddisonWesley.

Ramakrishnan, R., & Gehrke, J. (2002). Database Management Systems. McGraw-Hill.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.