

EXPLICACION PASO A PASO

tomamos todas las ventas, las consolidamos y creamos una nueva tabla no normalizada, después de transformar redujimos la cantidad de registros y se consolido el total de ordenes

- Extraemos la información del encabezado del pedido con los detalles del pedido
- Creamos una tabla temporal no normalizada (mostraba datos repetidos)

```
SELECT  
  
    s.SalesOrderID,  
  
    h.OrderDate,  
  
    c.CustomerID,  
  
    p.ProductID,  
  
    p.Name AS ProductName,  
  
    s.OrderQty,  
|  
    s.LineTotal  
  
INTO #SalesData  
  
FROM  
  
    AdventureWorks2019.Sales.SalesOrderDetail s  
  
    JOIN AdventureWorks2019.Sales.SalesOrderHeader h ON s.SalesOrderID = h.SalesOrderID  
  
    JOIN AdventureWorks2019.Sales.Customer c ON h.CustomerID = c.CustomerID  
  
    JOIN AdventureWorks2019.Production.Product p ON s.ProductID = p.ProductID;
```

Otro ejemplo

```

--Extraemos toda la información relevante sobre pedidos, productos, categorías, empleados y detalles del pedido.
--Creamos una tabla temporal llamada #TempSalesData para almacenar estos datos sin consolidar.
SELECT
    p.ProductID,
    p.Name AS ProductName,
    pc.Name AS CategoryName,
    e.BusinessEntityID AS EmployeeID,
    per.FirstName + ' ' + per.LastName AS EmployeeName,
    soh.SalesOrderID,
    soh.OrderDate,
    sod.OrderQty,
    sod.UnitPrice,
    sod.LineTotal
INTO #TempSalesData
FROM
    Sales.SalesOrderHeader AS soh
    INNER JOIN Sales.SalesOrderDetail AS sod ON soh.SalesOrderID = sod.SalesOrderID
    INNER JOIN Production.Product AS p ON sod.ProductID = p.ProductID
    INNER JOIN Production.ProductCategory AS pc ON p.ProductSubcategoryID = pc.ProductCategoryID
    INNER JOIN Sales.SalesPerson AS sp ON soh.SalesPersonID = sp.BusinessEntityID
    INNER JOIN HumanResources.Employee AS e ON sp.BusinessEntityID = e.BusinessEntityID
    INNER JOIN Person.Person AS per ON e.BusinessEntityID = per.BusinessEntityID;

```

- Consolidamos a nivel de cliente y tenemos un solo registro por cliente
- Resumimos el total, el número de pedido y la última compra

---Transformar datos

```

SELECT
    CustomerID,
    sum(LineTotal) AS TotalSpent,
    COUNT(DISTINCT SalesOrderID) AS OrderCount,
    MAX(OrderDate) AS LastOrderDate
into #TransformedSalesData
from
    #SalesData
group by
    CustomerID;

```

Otro Ejemplo

```

--Agrupamos los datos por empleado y categoría.
--Calculamos el número total de pedidos, la suma de ventas y la fecha del último pedido para cada combinación de empleado y categoría.
--Creamos otra tabla temporal, #ConsolidatedSalesData, con estos datos resumidos.
SELECT
    EmployeeID,
    EmployeeName,
    CategoryName,
    COUNT(DISTINCT SalesOrderID) AS NumberOfOrders,
    SUM(LineTotal) AS TotalSales,
    MAX(OrderDate) AS LastPurchaseDate
INTO #ConsolidatedSalesData
FROM
    #TempSalesData
GROUP BY
    EmployeeID,
    EmployeeName,
    CategoryName;

```

- De la tabla "TransformedSalesData nos pasamos a la tabla destino
- Creamos tabla en el data warehouse
- Cargamos los datos transformados en la tabla destino

```
--crea tabla en el data warehouse

USE AdventureWorksDW2019
-- Crear la tabla destino si no existe
IF NOT EXISTS (SELECT * FROM AdventureWorksDW2019.INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'CustomerSalesSummary')
BEGIN
    CREATE TABLE AdventureWorksDW2019.dbo.CustomerSalesSummary (
        CustomerID INT PRIMARY KEY,
        TotalSpent MONEY,
        OrderCount INT,
        LastOrderDate DATETIME
    );
END

USE AdventureWorksDW2019
-- Cargar los datos transformados en la tabla destino
INSERT INTO AdventureWorksDW2019.dbo.CustomerSalesSummary (CustomerID, TotalSpent, OrderCount, LastOrderDate)
SELECT
    CustomerID,
    TotalSpent,
    OrderCount,
    LastOrderDate
FROM
    #TransformedSalesData;
```

Otro Ejemplo

```
--Definimos la estructura de una tabla física llamada TablaDestino para almacenar los datos consolidados de forma permanente.
CREATE TABLE TablaDestino (
    EmployeeID INT,
    EmployeeName NVARCHAR(100),
    CategoryName NVARCHAR(100),
    NumberOfOrders INT,
    TotalSales DECIMAL(18, 2),
    LastPurchaseDate DATE
);

--Transferimos los datos consolidados desde la tabla temporal #ConsolidatedSalesData a la tabla física TablaDestino.
INSERT INTO TablaDestino (EmployeeID, EmployeeName, CategoryName, NumberOfOrders, TotalSales, LastPurchaseDate)
SELECT
    EmployeeID,
    EmployeeName,
    CategoryName,
    NumberOfOrders,
    TotalSales,
    LastPurchaseDate
FROM
    #ConsolidatedSalesData;
```