Northwind

Table of contents

1 Analisis de la base de datos Northwind

Introducción La base de datos Northwind es un conjunto de datos para prácticas de SQL, que simula una empresa de venta de productos alimenticios. Contiene información sobre clientes, empleados, pedidos, productos y proveedores. En este miniproyecto, realizaremos un análisis completo que incluye:

-

Creación de un diagrama entidad-relación

-

Consultas SQL avanzadas

-

Procedimientos almacenados

-

Funciones Triggers

-

Visualización de datos

2 Modificación de la Tabla Products: Atributos Dinámicos con JSON

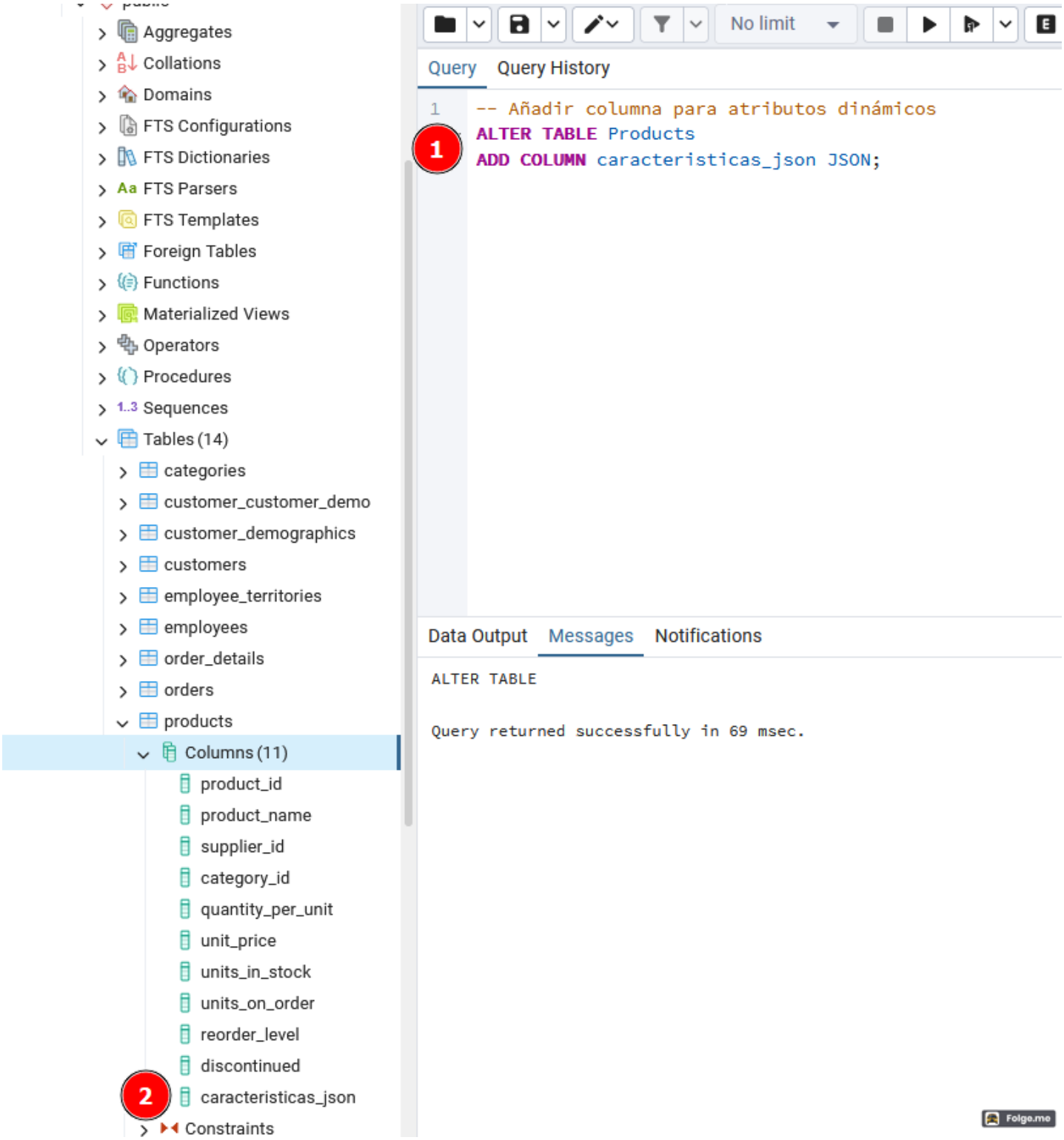
Añadir columna JSON a la tabla Products:

- Añadir columna para atributos dinámicos

```
-- Añadir columna para atributos dinámicos
ALTER TABLE Products
ADD COLUMN caracteristicas_json JSON;
```

- Verificar la estructura actualizada

```
DESCRIBE Products;
\d Productos:
```


Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (14)
categories
customer_customer_demo
customer_demographics
customers
employee_territories
employees
order_details
orders
products
Columns (11)
product_id
product_name
supplier_id
category_id
quantity_per_unit
unit_price
units_in_stock
units_on_order
reorder_level
discontinued
caracteristicas_json
Constraints
No limit
Query
Query History
1
-- Añadir columna para atributos dinámicos
ALTER TABLE Products
ADD COLUMN caracteristicas_json JSON;
1
2
Data Output
Messages
Notifications
ALTER TABLE
Query returned successfully in 69 msec.
Folge.me

3 Relleno de datos JSON

Actualizar productos existentes con datos

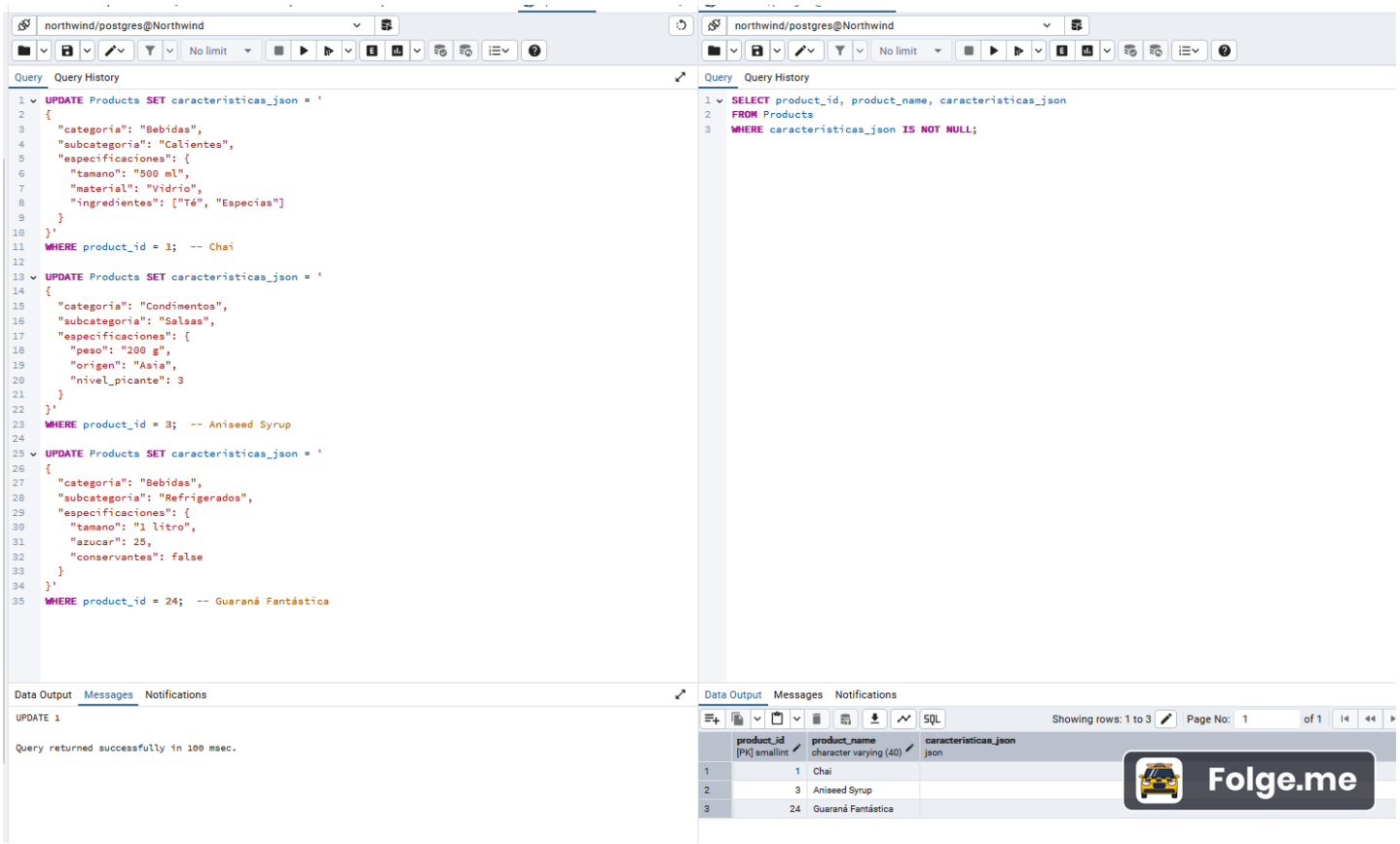
```
UPDATE Products SET caracteristicas_json = '
{
  "categoria": "Bebidas",
  "subcategoria": "Calientes",
  "especificaciones": {
    "tamano": "500 ml",
    "material": "Vidrio",
    "ingredientes": ["Té", "Especias"]
  }
}'
WHERE product_id = 1;  -- Chai

UPDATE Products SET caracteristicas_json = '
{
  "categoria": "Condimentos",
  "subcategoria": "Salsas",
  "especificaciones": {
    "peso": "200 g",
    "origen": "Asia",
    "nivel_picante": 3
  }
}'
WHERE product_id = 3;  -- Aniseed Syrup

UPDATE Products SET caracteristicas_json = '
{
  "categoria": "Bebidas",
  "subcategoria": "Refrigerados",
  "especificaciones": {
    "tamano": "1 litro",
    "azucar": 25,
    "conservantes": false
  }
}'
WHERE product_id = 24;  -- Guaraná Fantástica
```

Verificar datos actualizados

```
SELECT product_id , product_name, caracteristicas_json
FROM Products
WHERE caracteristicas_json IS NOT NULL;
```



4 Consultas a la columna JSON

```
SELECT caracteristicas_json->>'categoria' AS Categoria,
caracteristicas_json->>'subcategoria' AS subcategoria,
caracteristicas_json->'especificaciones'->'tamano' AS especificaciones
FROM Products
WHERE caracteristicas_json IS NOT NULL;
```

Query | Query History

```
1  SELECT caracteristicas_json->>'categoria' AS Categoria,
2  caracteristicas_json->>'subcategoria' AS subcategoria,
3  caracteristicas_json->'especificaciones'->'tamano' AS especificaciones
4  FROM Products
5  WHERE caracteristicas_json IS NOT NULL;
```

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

	categoria text	subcategoria text	especificaciones json
1	Bebidas	Calientes	"500 ml"
2	Condimentos	Salsas	[null]
3	Bebidas	Refrigerados	"1 litro"

5 Ampliacion de Base de datos con Nueva Tabla

Descuentos por volumen

```
CREATE TABLE volume_discounts (
    DiscountID INT AUTO_INCREMENT PRIMARY KEY,
    ProductID INT NOT NULL,
    MinQuantity INT NOT NULL,
    Discount DECIMAL(4,2) NOT NULL CHECK (Discount BETWEEN 0 AND 1),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

ery Query History

```
CREATE TABLE volume_discounts (
    discount_id INT PRIMARY KEY,
    product_id INT NOT NULL,
    MinQuantity INT NOT NULL,
    Discount DECIMAL(4,2) NOT NULL CHECK (Discount BETWEEN 0 AND 1),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
INSERT INTO volume_discounts (discount_id,product_id, MinQuantity, Discount) VALUES
-- Descuentos para Chai (ProductID = 1)
(1,1, 10, 0.05),    -- 5% de descuento para compras de 10+ unidades
(2,1, 20, 0.10),    -- 10% de descuento para compras de 20+ unidades
(3,1, 50, 0.15),    -- 15% de descuento para compras de 50+ unidades

-- Descuentos para Chang (ProductID = 2)
(4,2, 5, 0.07),     -- 7% de descuento para compras de 5+ unidades
(5,2, 15, 0.12),    -- 12% de descuento para compras de 15+ unidades

-- Descuentos para Sir Rodney's Marmalade (ProductID = 20)
(6,20, 8, 0.10),    -- 10% de descuento para compras de 8+ unidades
(7,20, 30, 0.25),   -- 25% de descuento para compras de 30+ unidades

-- Descuentos para Louisiana Hot Spiced Okra (ProductID = 46)
(8,46, 12, 0.15),   -- 15% de descuento para compras de 12+ unidades
(9,46, 24, 0.20),   -- 20% de descuento para compras de 24+ unidades

-- Descuentos para Scottish Longbreads (ProductID = 59)
(10,59, 6, 0.05),    -- 5% de descuento para compras de 6+ unidades
(11,59, 12, 0.12),  -- 12% de descuento para compras de 12+ unidades
(12,59, 36, 0.30);  -- 30% de descuento para compras de 36+ unidades
```

6 Creacion de vistas

Analisis de volumen de descuento

```
CREATE VIEW analisis_volumen_descuento AS
SELECT
    vd.discount_id,
    vd.product_id,
    p.product_name,
    p.unit_price,
    vd.MinQuantity,
    vd.Discount AS DiscountRate,
    -- Precio con descuento unitario
    ROUND((p.unit_price * (1 - vd.Discount))::NUMERIC, 2) AS DiscountedPrice,
    -- Ahorro por unidad
    ROUND((p.unit_price * vd.Discount)::NUMERIC, 2) AS SavingsPerUnit,
    -- Precio mínimo para el lote
    (vd.MinQuantity * p.unit_price) AS OriginalBatchPrice,
    -- Precio con descuento para el lote
    ROUND((vd.MinQuantity * p.unit_price * (1 - vd.Discount))::NUMERIC, 2) AS DiscountedBatchPrice,
    -- Ahorro total en el lote
    ROUND((vd.MinQuantity * p.unit_price * vd.Discount)::NUMERIC, 2) AS TotalSavings
FROM volume_discounts vd
JOIN Products p ON vd.product_id = p.product_id;
```