

Optimización de Consulta SQL: Clientes con más órdenes por territorio

Consulta Original Analizada

La consulta original busca listar los nombres de los clientes con más órdenes en cada territorio. Utiliza una subconsulta con `row_number()` para rankear a los clientes por cantidad de compras en cada territorio, y luego filtra solo los primeros (rn = 1).

Fortalezas de la consulta original:

- Correctamente utiliza `row_number()` para identificar los clientes top por territorio
- Buen uso de JOINS para relacionar las tablas necesarias
- Agrupación adecuada por los campos relevantes

Debilidades/Oportunidades de mejora:

1. ****Nombres de alias****: Usa T1, T3, T4 que no son descriptivos
2. **Subconsulta innecesaria**: Podría convertirse en una CTE para mejor legibilidad
3. **Filtrado**: El WHERE rn=1 podría ser más eficiente con RANK() o FIRST_VALUE()
4. **Ordenamiento**: Falta ORDER BY en el resultado final

Consulta Optimizada

WITH ClientesPorTerritorio AS (

SELECT

soh.TerritoryID,

p.FirstName,

p.LastName,

COUNT(*) AS Compras_realizadas,

RANK() OVER (PARTITION BY soh.TerritoryID ORDER BY COUNT(*) DESC) AS ranking

FROM SalesOrderHeader soh

INNER JOIN Customer c ON soh.CustomerID = c.CustomerID

INNER JOIN Person p ON c.PersonID = p.BusinessEntityID

GROUP BY soh.TerritoryID, c.CustomerID, p.FirstName, p.LastName

)

Sánchez Herrera Armando Eduardo. Tarea 04. Optimización de la consulta 1
Práctica 2.

SELECT

TerritoryID,

FirstName,

LastName,

Compras_realizadas

FROM ClientesPorTerritorio

WHERE ranking = 1

ORDER BY TerritoryID;

Mejoras Implementadas

1. CTE en lugar de subconsulta:

- Más legible y mantenible
- Mejor organización del código

2. Alias descriptivos:

- Reemplazo de T1, T3, T4 por soh (SalesOrderHeader), c (Customer), p (Person)

3. RANK() en lugar de ROW_NUMBER():

- Maneja mejor los empates (si dos clientes tienen el mismo número máximo de compras, ambos aparecerán)
- Más semánticamente correcto para este caso

4. ORDER BY agregado:

- Resultados ordenados por TerritoryID para mejor presentación

5. Estructura más clara:

- Separación lógica de la preparación de datos y el filtrado final

Consideraciones adicionales

1. Si hay territorios sin órdenes, no aparecerán en los resultados. Si esto es importante, habría que usar LEFT JOINS.

2. Para bases de datos muy grandes, podrían considerarse índices en:

- SalesOrderHeader(TerritoryID, CustomerID)
- Customer(CustomerID, PersonID)

Sánchez Herrera Armando Eduardo. Tarea 04. Optimizacion de la consulta 1
Practica 2.

- Person(BusinessEntityID)

3. Si el rendimiento sigue siendo un problema, podría evaluarse una solución con FIRST_VALUE() o incluso una aproximación con TOP WITH TIES.