



# TAREA#3

## Bases de Datos Distribuidas

Descripción breve

REPORTE DE TAREA 3

- Urrutia González Brenda
  - Olea García Alan
- Juárez Anguiano Mario Alexis

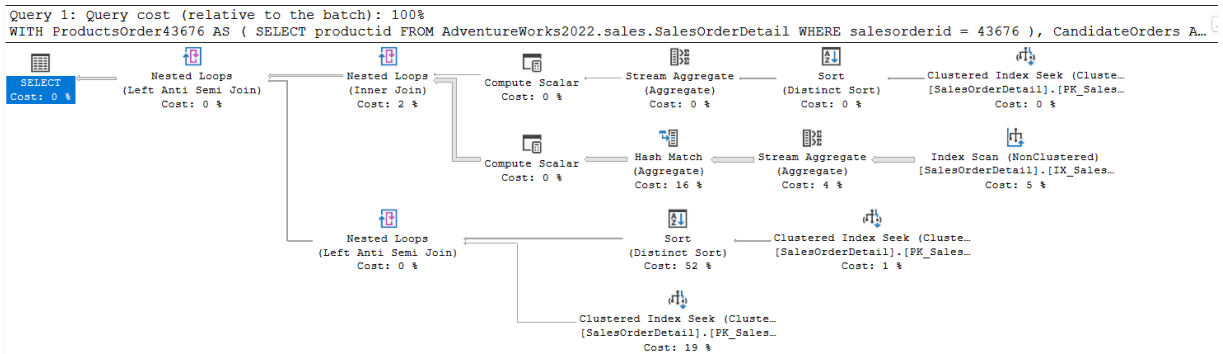
Equipo 03

03/06/2025

## CONSULTA 3 DADA POR EL PROFESOR

```
WITH ProductsOrder43676 AS (  
    SELECT productid  
    FROM AdventureWorks.sales.SalesOrderDetail  
    WHERE salesorderid = 43676  
),  
CandidateOrders AS (  
    SELECT salesorderid, productid  
    FROM AdventureWorks.sales.SalesOrderDetail  
    WHERE salesorderid <> 43676  
)  
SELECT salesorderid  
FROM CandidateOrders  
GROUP BY salesorderid  
HAVING COUNT(DISTINCT productid) >= (SELECT COUNT(DISTINCT productid)  
    FROM ProductsOrder43676)  
AND NOT EXISTS (  
    SELECT productid  
    FROM ProductsOrder43676  
EXCEPT  
    SELECT productid  
    FROM CandidateOrders c  
    WHERE c.salesorderid = CandidateOrders.salesorderid  
);
```

## PLAN DE EJECUCIÓN



## CONSULTA 3 DE LA PRÁCTICA

USE AdventureWorks2022;

GO

SELECT DISTINCT Salesorderid

FROM Sales.SalesOrderDetail AS OD

WHERE NOT EXISTS

(

SELECT \*

FROM (SELECT productid

from Sales.SalesOrderDetail

where salesorderid=43676) as P

WHERE NOT EXISTS

(

SELECT \*

FROM Sales.SalesOrderDetail

WHERE OD.salesorderid =

AND (OD2.productid =

)

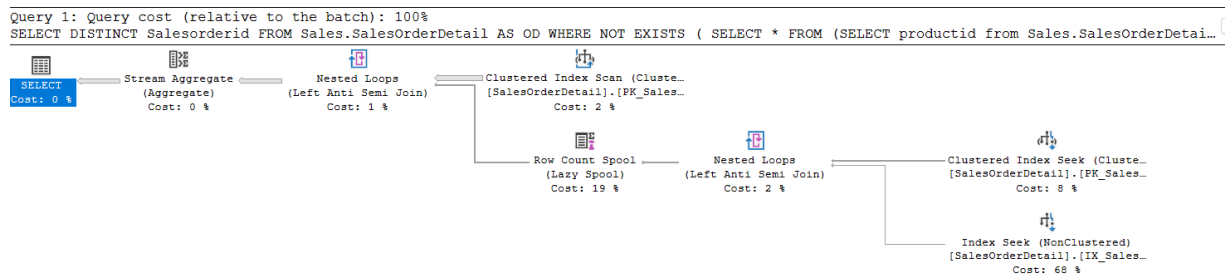
AS OD2

OD2.salesorderid

P.productid)

);

## PLAN DE EJECUCIÓN



## COMPARACIÓN ENTRE PLANES DE EJECUCIÓN

<i>Aspecto</i>	<b>Primer plan de ejecución</b>	<b>Segundo plan de ejecución</b>
<i>Estructura general</i>	Usa CTEs para definir conjuntos (productos de orden 43676 y productos de órdenes candidatas). Luego agrupa y filtra con EXCEPT para verificar inclusión.	Consulta simple con doble NOT EXISTS anidado para verificar que no falten productos de la orden 43676 en cada orden candidata.
<i>Obtención productos base</i>	CTE ProductsOrder43676 materializa productos de orden 43676 solo una vez.	Subconsulta interna obtiene productos de la orden 43676, se evalúa dentro del filtro NOT EXISTS.
<i>Filtrado candidatos</i>	Agrupa productos por orden y filtra órdenes con al menos tantos productos distintos como la orden base, para reducir candidatos antes de la comparación final.	No hace agrupación previa; evalúa cada orden directamente con el filtro doble NOT EXISTS sin prefiltrar por número de productos.
<i>Verificación de inclusión</i>	Usa la operación EXCEPT para encontrar productos faltantes en cada orden candidata; si no hay, pasa el filtro NOT EXISTS.	Usa doble NOT EXISTS para verificar para cada producto de la orden base si existe en la orden candidata. Si falta alguno, se descarta.
<i>Complejidad computacional</i>	Agrupación y conteo COUNT(DISTINCT) ayudan a filtrar órdenes candidatas, lo que puede reducir el número de evaluaciones EXCEPT. Sin embargo, EXCEPT puede ser costoso.	Sin agrupación previa, la doble evaluación anidada puede ser menos eficiente, ya que se verifica producto por producto para cada orden.
<i>Uso de índices</i>	Puede beneficiarse de índices para filtrado, agrupación y para acelerar la operación EXCEPT.	Puede beneficiarse de índices compuestos (salesorderid, productid) para acelerar las búsquedas en las subconsultas anidadas.
<i>Legibilidad y mantenibilidad</i>	Más estructurada con CTEs y lógica clara, fácil de entender y mantener para conjuntos complejos.	Más compacta pero puede ser menos intuitiva por la anidación de NOT EXISTS.
<i>Posible rendimiento</i>	Más eficiente en escenarios con muchas órdenes y productos, gracias a la prefiltración por conteo y uso de EXCEPT como operación set.	Podría ser menos eficiente en grandes volúmenes porque hace chequeo producto a producto para cada orden sin prefiltrado.
<i>Resultado final</i>	Ambas devuelven órdenes que contienen todos los productos de la orden 43676.	Ambas devuelven órdenes que contienen todos los productos de la orden 43676.

## CONCLUSIÓN

La comparación entre ambos planes de ejecución muestra que, si bien ambos devuelven el mismo resultado, el primer enfoque con CTEs y el uso de EXCEPT tiene una mejor capacidad para manejar conjuntos de datos grandes debido a su filtrado previo. Por otro lado, el segundo enfoque, aunque más sencillo, es menos eficiente en escenarios con altos volúmenes de datos, ya que requiere comparar cada producto individualmente.

En resumen, el primer plan es más eficiente y escalable en situaciones con grandes bases de datos, mientras que el segundo es adecuado para conjuntos de datos más pequeños o cuando se requiere una consulta más compacta, aunque puede resultar menos eficiente en el procesamiento de grandes volúmenes.