

# TAREA#3

## Bases de Datos Distribuidas

Descripción breve

REPORTE DE PRÁCTICA 3

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

Equipo 03  
03/06/2025

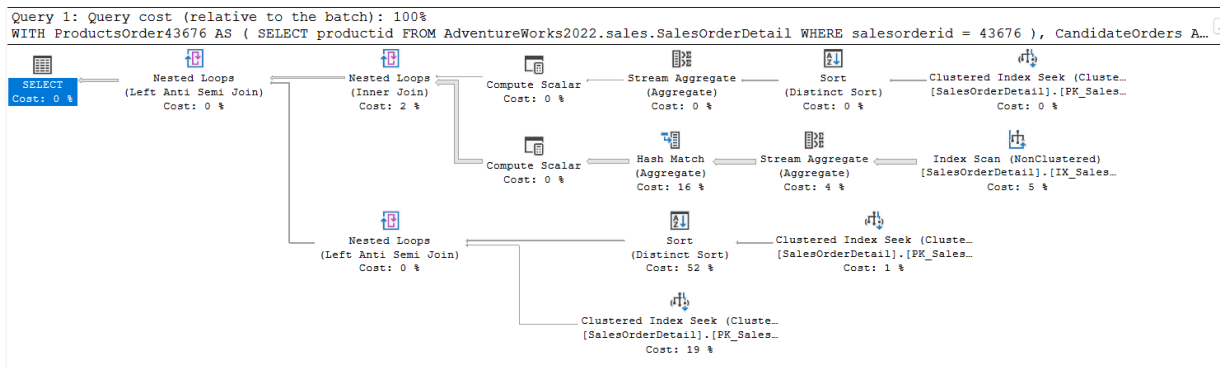
## Analizar el plan de ejecución de la consulta 3 de la práctica PLANES DE EJECUCIÓN con la siguiente solución

```

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



1. Evaluación del CTE ProductsOrder43676:
  - Se ejecuta primero.
  - Extrae y guarda en memoria o en un objeto temporal los productid de la orden 43676.
  - Generalmente, esto se hace mediante un índice en salesorderid para acceder rápido.
2. Evaluación del CTE CandidateOrders:
  - Se ejecuta después.
  - Se obtiene la tabla con todas las combinaciones salesorderid, productid de las órdenes distintas a la 43676.
  - Si hay índice por salesorderid, la consulta será más eficiente.
3. Agrupación y filtrado:
  - La consulta principal agrupa los datos por salesorderid.
  - El motor calcula COUNT(DISTINCT productid) para cada grupo.
  - Compara con el número de productos distintos de la orden 43676 (resultado del subconsulta).
  - Este filtro reduce el número de candidatos.
4. Evaluación del predicado NOT EXISTS (...):
  - Para cada grupo (cada orden candidata), se ejecuta el subconsulta para verificar que no exista ningún producto en ProductsOrder43676 que falte en esa orden.
  - La subconsulta:
    - Obtiene la diferencia entre los productos de la orden 43676 y los productos de la orden candidata usando EXCEPT.
    - Si el resultado es vacío, quiere decir que la orden candidata contiene todos los productos de la orden 43676.

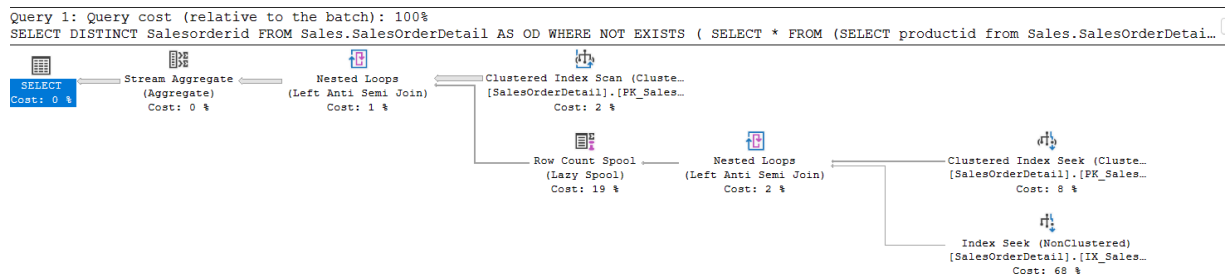
## -CONSULTA 3 DE PRÁCTICA

```
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 =
                OD2.salesorderid
                AND (OD2.productid =
                P.productid)
            )
    );
```

## -Plan de ejecución



1. Escaneo o index seek en SalesOrderDetail para cada orden distinta:

- Se recorren las órdenes distintas con sus productos.
- Para cada orden, se realiza la verificación del segundo NOT EXISTS.

## 2. Verificación anidada:

- Para cada producto de la orden 43676, se busca si ese producto existe en la orden actual (OD.salesorderid).
- Esto puede hacerse con índices en (salesorderid, productid) para acelerar la búsqueda.

## 3. Filtrado:

- Se descartan las órdenes que no cumplen con que todos los productos de la orden 43676 estén presentes.

## 4. Selección de resultados:

- Se devuelve la lista distinta de salesorderid que cumplen con la condición.

## 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>	Agrupar 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.