EJERCICIO 2

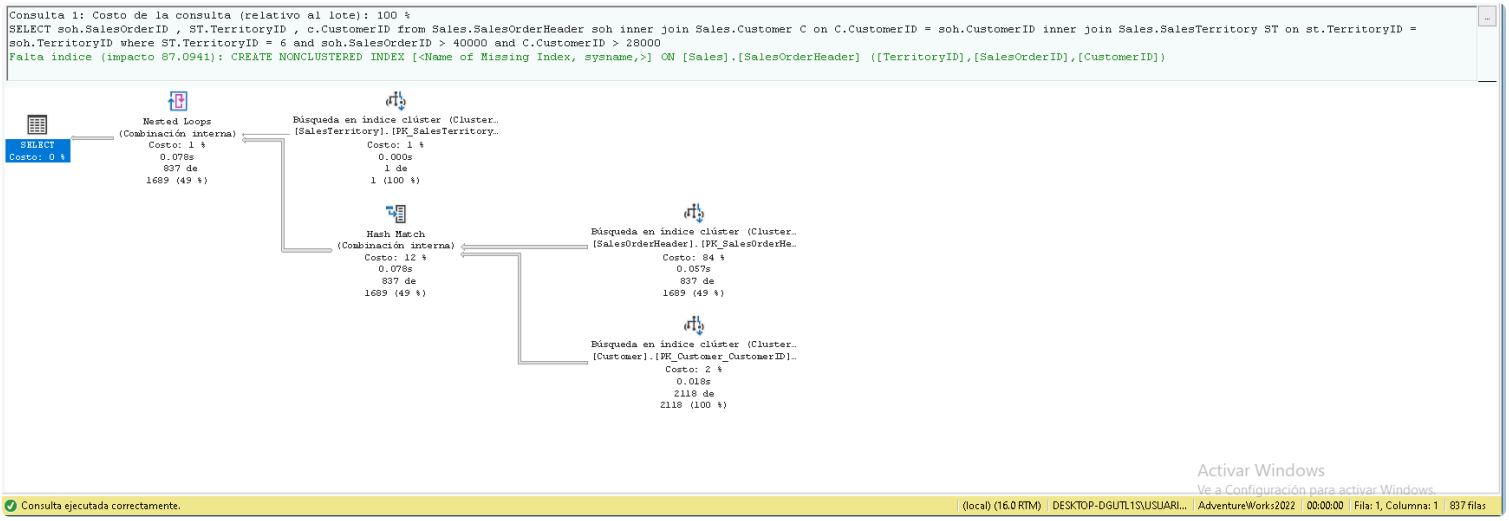
Consulta

```
SELECT soh.SalesOrderID , ST.TerritoryID , c.CustomerID from Sales.SalesOrderHeader soh
inner join Sales.Customer C
on C.CustomerID = soh.CustomerID
inner join Sales.SalesTerritory ST
on st.TerritoryID = soh.TerritoryID
where  ST.TerritoryID = 6 and  soh.SalesOrderID > 40000 and C.CustomerID > 28000
```

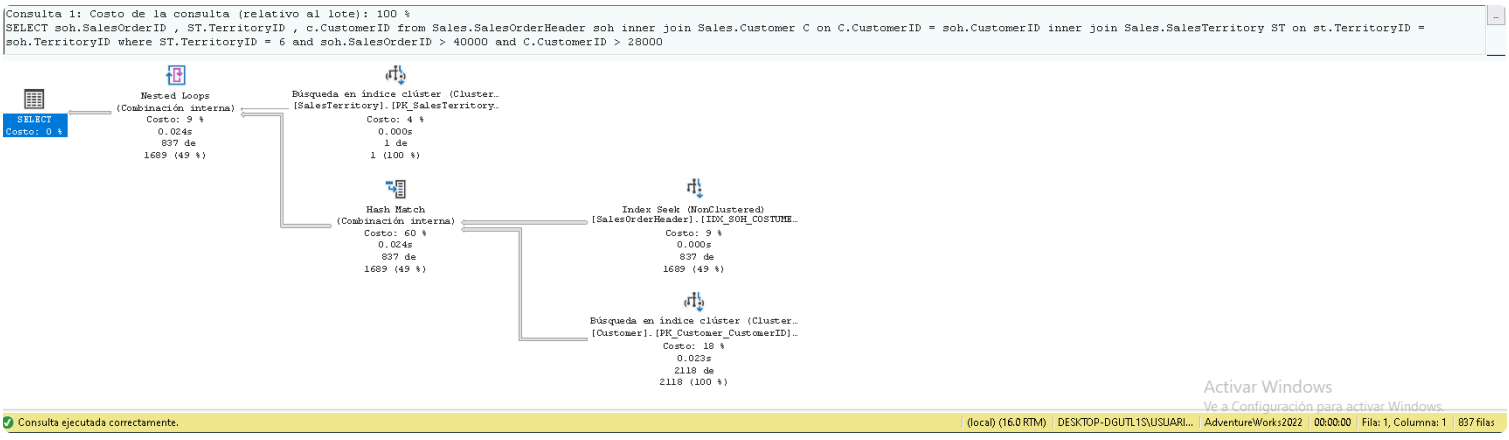
Es una consulta simple donde se quiere saber cuales son las ordenes mayores de 40000 , del `consumidorid` mayor de 28000 y que hayan sido en el `territorioid` = 6

La consulta ya en si es rápida pero con la ayuda de los índices se puede conseguir en menos tiempo.

sin índices



con índice



Creación del índice

```
create nonclustered index IDX_SOH_COSTUMER_TERRITORY
on Sales.SalesOrderHeader (TerritoryId,CustomerId,SalesOrderId)
```

Cree una conexión entre estos 3 atributos de la tabla SALESORDERHEADER para que puedan ser filtrados en menos tiempo en la condición "WHERE"

Mejoras tras la inclusión de este índice

Realmente , si se logra apreciar en las capturas de imagen , cada sección del plan ejecución se ha visto mejorada en el tiempo de respuesta. Aunque haya sido mínimo el tiempo de mejora que rondo en milisegundos.

Donde si se ha visto mucho cambio a sido en el Costo de CPU porque cuando no tienes el índice creado usa la búsqueda de índices `Cluster` , al usar un `JOIN` para tabla `salesorderheader` llamara a todas las filas de esa tabla y es por eso que cuando ves en el plan de ejecución el numero de filas leídas en la búsqueda de índices `cluster` de esta tabla sale `31465` filas leídas y en cambio cuando el índice `nonclustered` existe entre `TERRITORIID , SALESORDERID Y CUSTOMERID` el numero de filas leídas se reduce a `847`