METODOS DE JOIN

OPERACIONES

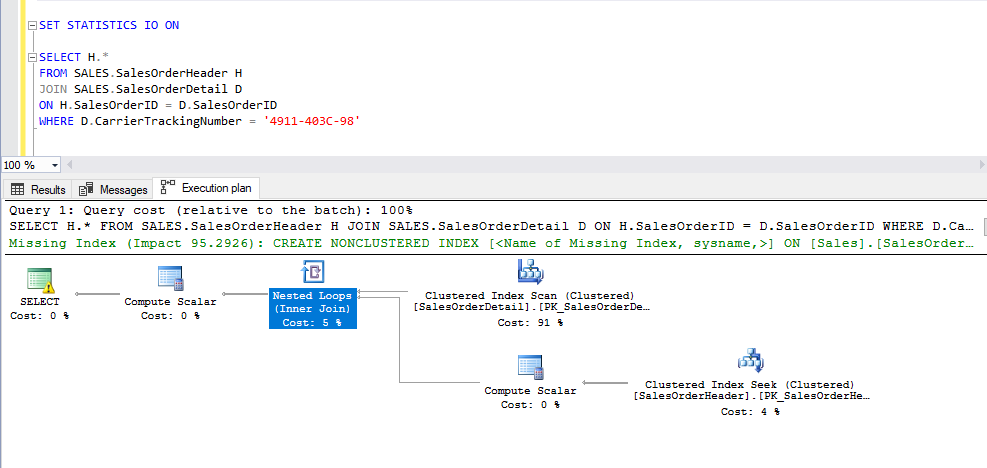
NESTED LOOPS

MERGE JOIN

HASH JOIN

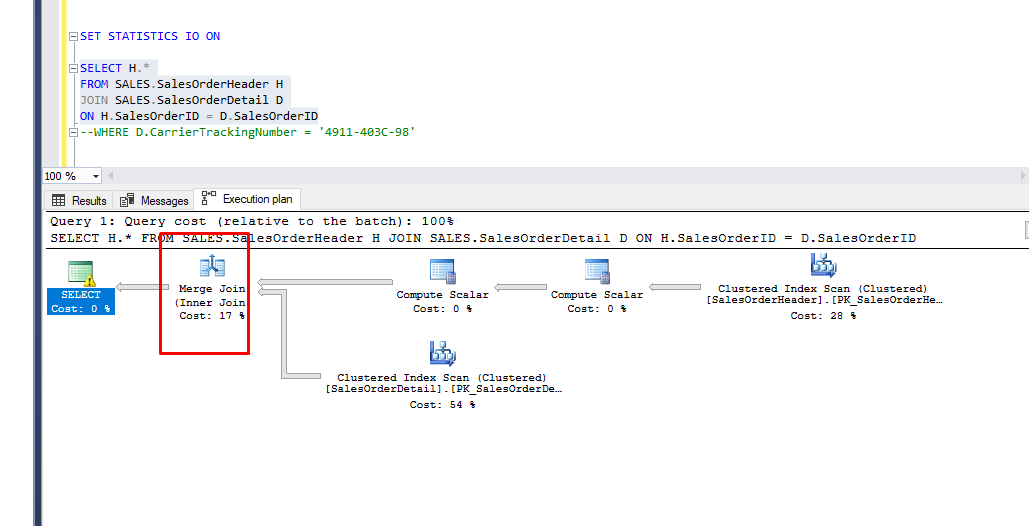
NESTED LOOPS

Media volumenes de datos.



Va a tomar el orden de la cabecera la salesorderheader y luego por cada ítem busca en la otra tabla.

El optimizador lógicamente resuelve que pares de tablas van joineando y el resultado se va joinendo con el par siguiente



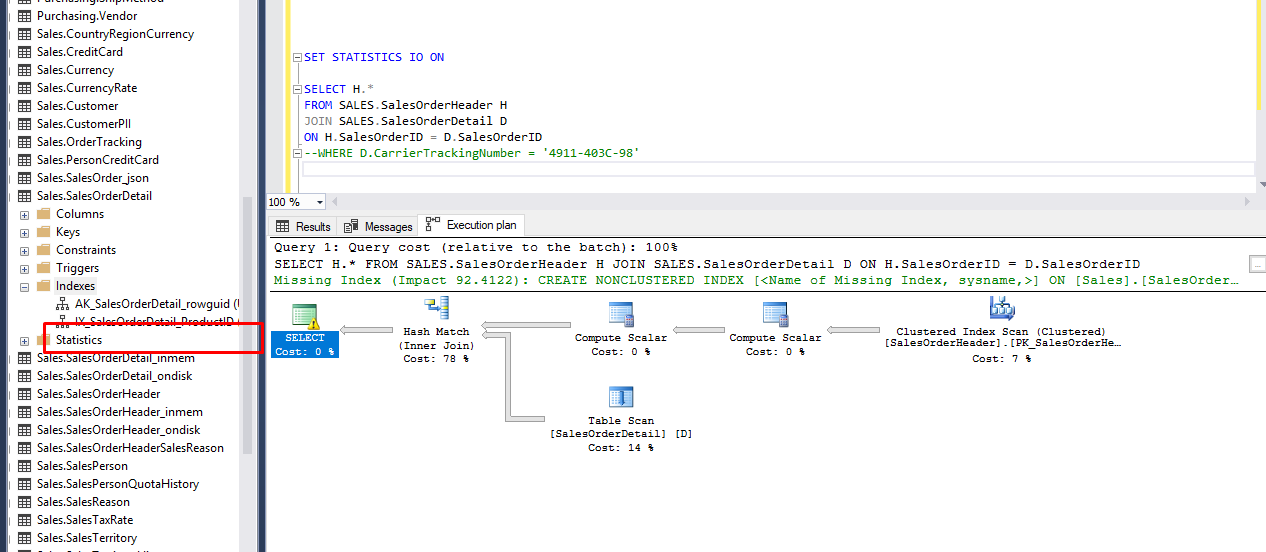
MERGE JOIN

Sin predicado la operación que utiliza es un merge Join , la ventaja es que no puede tomar el loop menos costoso porque no tiene la referencia, tiene que barrerlos todos, al estar indexados están ordenados, siempre la clave siguiente va a estar ordenado no tiene que volver a organizar las búsquedas.

Merge contiene ambos que están ordenados y volúmenes muy grande en ambos casos.

Que pasa si hay uno que esta ordenado y la otra no, entonces para hacer el marge va a aparecer una operación

Implícita de shorting, va a ordenar el otro dataset y luego va a aplicar la operación de merge. Para eso veo el plan de ejecución para saber si esta realizando la reordenación de un dataset lo cual es costoso.

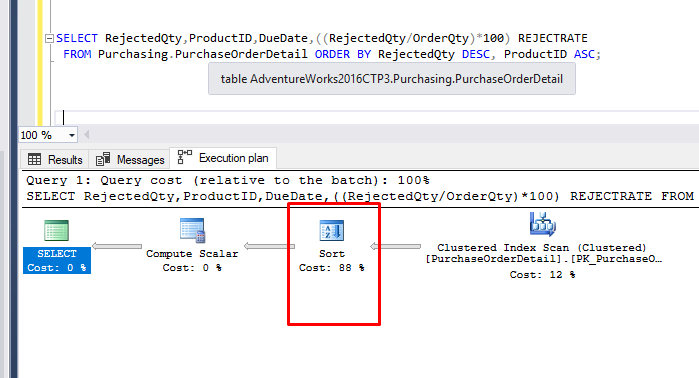


Eliminando el de la detail hace el siguiente plan

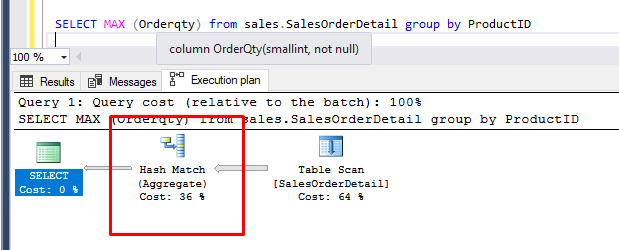
HASH JOIN

Cuando tenemos dos dataset muy grandes y no tenemos indices, entonces aplica la función de hash, carga un valor y devuelve la misma clave, crea una tabla de hash donde va armando referencia a las filas que corresponden a ese hash, es mejor que hacer un sorting, minimiza el nuemero de escaneaos, se asegura que el escaneo de un input lo va a hacer una vez para armar al hash table.

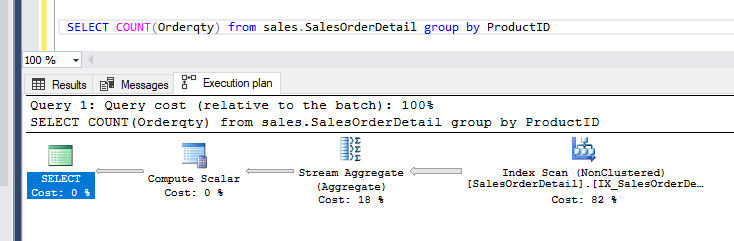
La operación de sort es la que mas consume. Hay que mejorar el order by creando un índice contemplando las dos columnas de ordenamiento



Agrupamiento, el conjunto de datos no queda ordenado por eso usa el hash match y no utiliza el cluster index



En la siguiente si utiliza el noncluster, entonces cambia el algoritmo y va por el stream aggregate



UNIO VS UNION ALL

En términos de performance es mejor el Union all porque no hace la operación de distinct.

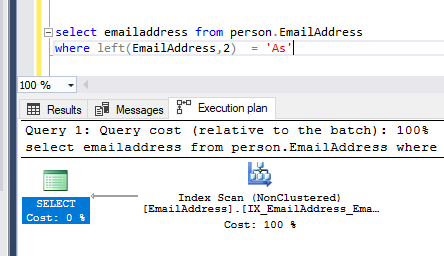
Query = (subconsulta) lo mejor seria guardar ese valor en una variable y luego hacer el igual por esa.

Exist donde encuentra el valor deja de contar.

Evitar

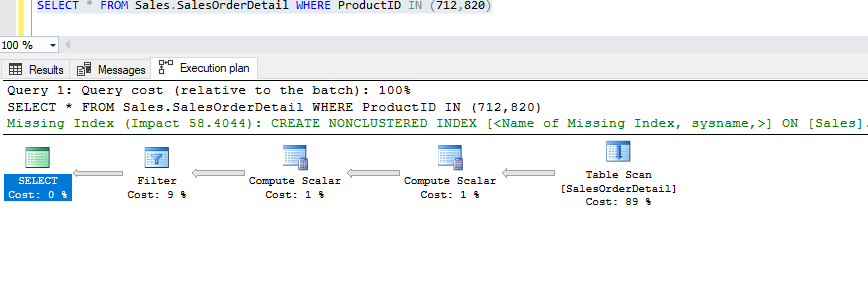
Where substr (firstname,1,1) = ‘r’ 🡪 aca no puede utilizer si lo hubiera un indice por la columna, porque esta el subtr

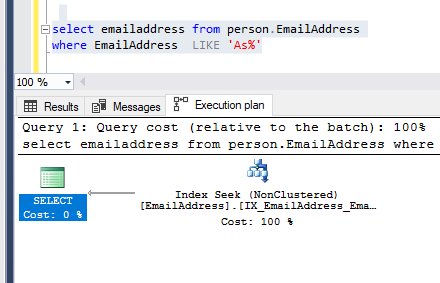
Where firstname like ‘r%’ 🡪 aca si puede utilizarlo. “Desenmascarar la columna”

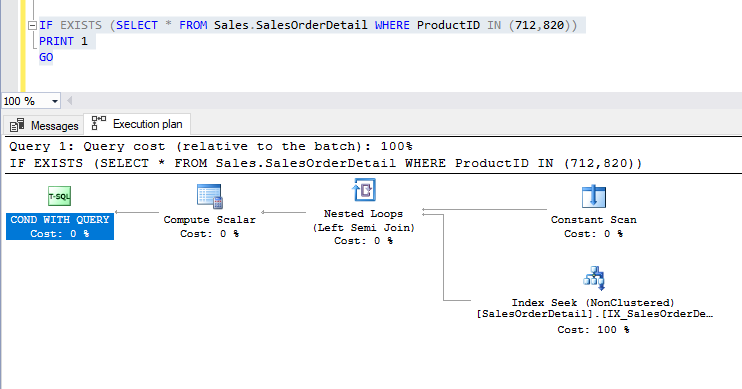


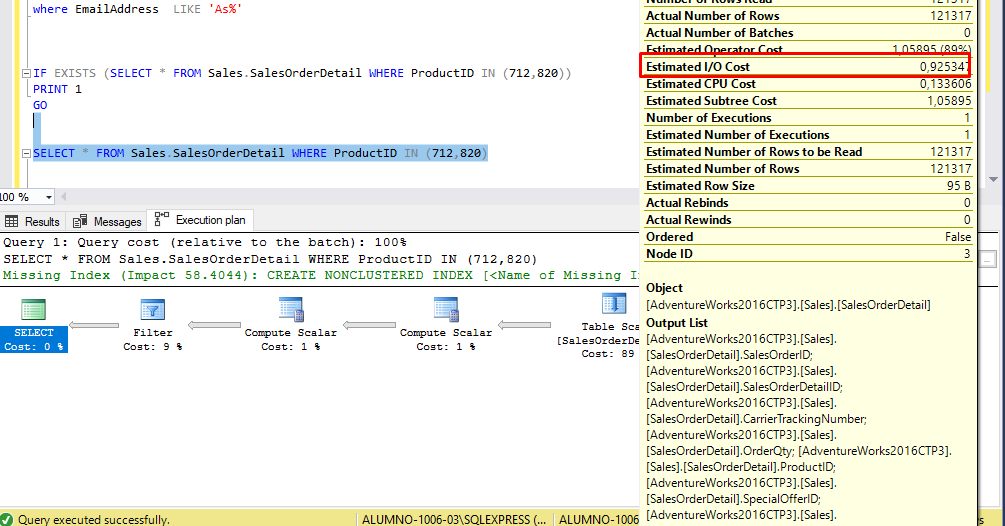
que es calificar la columna?

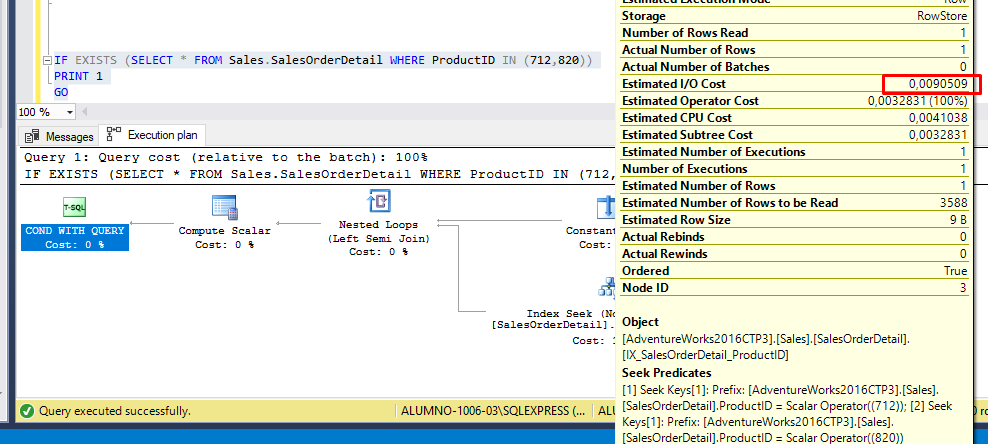
tabla.columna 🡪 para no ir a buscarlo al diccionario de datos.











PARAMETER SNIFFING

Recompile (se genera siempre el plan de ejecucion)