

# **Plan de Ejecución**

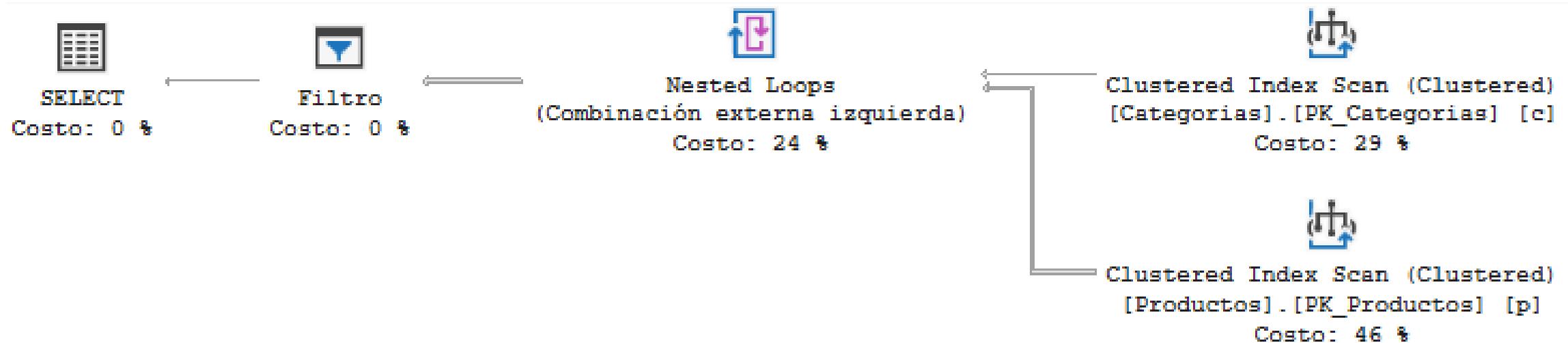
Antonio Espín Herranz

# **Control + L (consulta seleccionada)**

- Menú consulta → mostrar el plan de ejecución estimado.
- Buscar categorías que no tienen productos:  
select c.[NombreCategoría]  
from Categorias c left join Productos p  
on c.[IdCategoría] = p.[IdCategoría]  
where p.IdCategoría is null;

# Plan de Ejecución

- Categorías 8 filas y Productos 77 filas
- Si buscamos por clave primaria → utiliza los índices CLUSTERED más rápidos



# Secuencia

- Lee toda la tabla de Categorías
  - Utiliza el index: CLUSTERED PK de Categorías
  - Estima 8 filas
- Lee toda la tabla de Productos:
  - Utiliza el index: CLUSTERED PK de Productos
  - Estima 77 filas
- Nested Loop Join:
  - Compara cada categoría con los productos por IdCategoria
- Aplica el filtro is null

# Plan de ejecución

## De categorías:

Costo	Operación física	Clustered Index Scan (Clustered)
= Clustered Index [Productos]. [PK Costo]	Operación lógica	Examen de índice agrupado
	Modo de ejecución estimado	Row
	Almacenamiento	RowStore
	Costo de E/S estimado	0,003125
	Costo de operador estimado	0,0032908 (29%)
	Costo de CPU estimado	0,0001658
	Costo de subárbol estimado	0,0032908
	Número de ejecuciones estimado	1
	Número estimado de filas para todas las ejecuciones	8
	Número estimado de filas por ejecución	8
	Número estimado de filas que se leerán	8
	Tamaño de fila estimado	30 B
	Ordenado	False
	Id. de nodo	2
<b>Objeto</b>		
[Empresa_PRC].[dbo].[Categorias].[PK_Categorias] [c]		
<b>Lista de salida</b>		
[Empresa_PRC].[dbo].[Categorias].IdCategoria; [Empresa_PRC].[dbo].[Categorias].NombreCategoria		

## De productos: $8 \times 77 = 616$

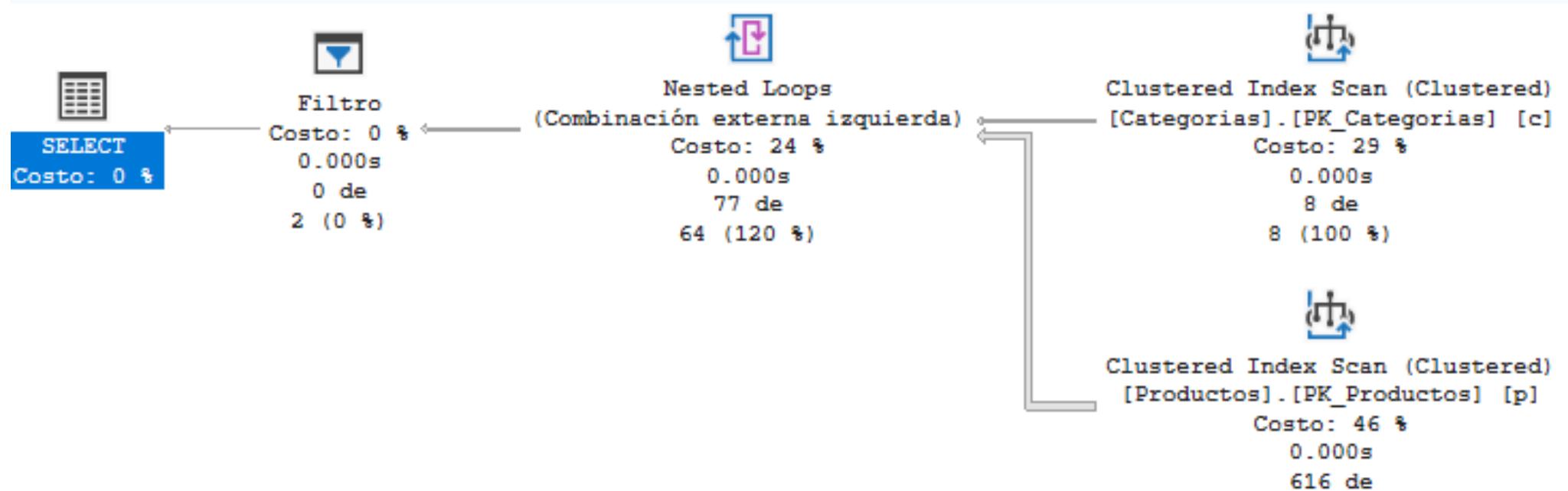
Costo	Operación física	Clustered Index Scan (Clustered)
	Operación lógica	Examen de índice agrupado
	Modo de ejecución estimado	Row
	Almacenamiento	RowStore
	Costo de E/S estimado	0,0039442
	Costo de operador estimado	0,0052498 (46%)
	Costo de CPU estimado	0,0001632
	Costo de subárbol estimado	0,0052498
	Número de ejecuciones estimado	8
	Número estimado de filas para todas las ejecuciones	616
	Número estimado de filas por ejecución	77
	Número estimado de filas que se leerán	77
	Tamaño de fila estimado	11 B
	Ordenado	False
	Id. de nodo	3
<b>Objeto</b>		
[Empresa_PRC].[dbo].[Productos].[PK_Productos] [p]		
<b>Lista de salida</b>		
[Empresa_PRC].[dbo].[Productos].IdCategoria		

# Plan de Ejecución

- Si hacemos un inner join, buscará la tabla más pequeña, y empieza por esta.
- Para ver la tabla que usa primero SQL Server , activar el plan de ejecución actual (control + M), una vez activado, ejecutar la consulta.
  - Ya pone el número de filas leído.

# Plan Ejecución actual

- Control + M (o activar icono)
- Menú consulta -> incluir el plan de ejecución real
- Ojo ejecutar la consulta para verlo.



# Con tablas grandes

- Plan estimado del actual será distinto.
- Podemos ver:
- **Hash Match gigante**
- SQL Server no puede usar índices para el join.
- **Sort costoso**
- Falta índice que ordene previamente.
- **Parallelism**
- SQL Server divide el trabajo entre CPUs.
- **Spills a tempdb**
- Falta memoria para hash o sort.
- **Scans enormes**
- Falta índice en la columna del join o del filtro.
- **Diferencias enormes entre estimado y real**
- Estadísticas desactualizadas o mala cardinalidad.
- **Memory Grant Warning**
- SQL Server pidió demasiada o muy poca memoria

# Revisar operadores

- **Table Scan**

- Recorre TODA la tabla
- Señal de que falta un índice

- **Index Scan**

- Recorre todo el índice
- Puede estar bien o ser un problema según el tamaño

- **Index Seek**

- Lo mejor SQL Server usa un índice de forma selectiva

- **Nested Loop**

- Bueno para pocas filas
- Malo si la tabla derecha es enorme

- **Hash Match**

- SQL Server no encontró un índice adecuado
- Consumir memoria
- Puede provocar spills a tempdb

- **Sort**

- Costoso si no hay índice que ya ordene

- **Key Lookup**

- SQL Server encuentra filas en un índice nonclustered
- Luego va a buscar columnas adicionales al clustered
- Si ocurre miles de veces → muy caro