

Optimización en Sql Server

- ♦ Juan Luís Carratalá Castillo
- ♦ Víctor Perdomo Pastor

Optimización en Sql Server

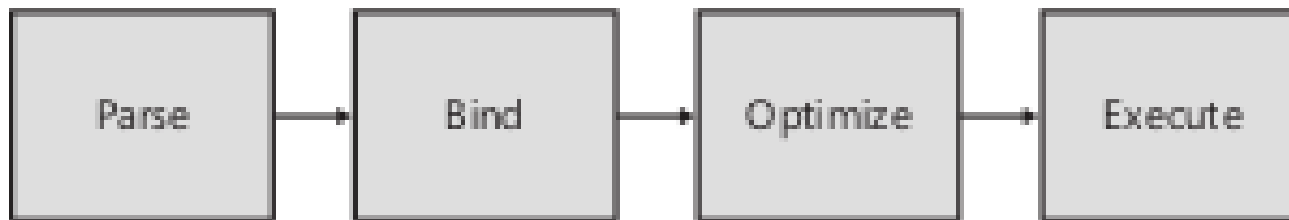
- El Optimizador de consultas
- Herramientas de diagnóstico
- Índices en SQL Server 2008

El Optimizador de consultas

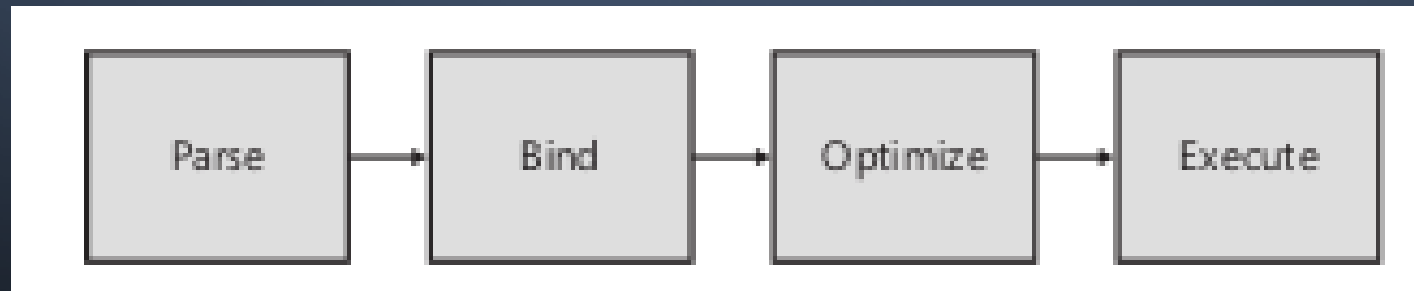
- ♦ ¿Qué es el Optimizador?
- ♦ Reglas
- ♦ Arquitectura
- ♦ Optimización de actualizaciones
- ♦ Estadísticas, estimación y coste

El Optimizador de consultas

- ¿Qué es el Optimizador?
- La ejecución de una consulta pasa cuatro fases.



El Optimizador de consultas



- ♦ Parse: La consulta se transforma en una representación en árbol.
- ♦ Bind: Se validan que datos, tablas y columnas.
- ♦ Optimize: Busca y evalúa distintos planes de ejecución.
- ♦ Execute: Ejecuta el plan seleccionado.

El Optimizador de consultas

- ♦ Sistema basado unicamente en costes.
- ♦ Busca planes de ejecución.
- ♦ Evalúa cada uno de ellos conforme los genera.
- ♦ Selecciona el que se acerque a los requisitos mínimos.

El Optimizador de consultas

- ◆ Reglas: Transformaciones del árbol en equivalencias.
- ◆ Substitución: Reescriben el árbol.
- ◆ Exploración: Equivalencias matemáticas.
- ◆ Implementación: Convierte en reales las transformaciones anteriores.

El Optimizador de consultas

- ♦ Propiedades: Información para toma de decisiones de optimización.
- ♦ Almacena información de sub-árboles.
- ♦ Claves primarias, restricciones, constantes,...

El Optimizador de consultas

- ♦ Ejemplo:

```
SELECT *  
FROM DomainTable D1  
INNER JOIN DomainTable D2  
ON D1.col1=D2.col1  
WHERE D1.col1 > 5 AND D2.col1 < 0;
```

El Optimizador de consultas

- ♦ Ejemplo:

El Optimizador de consultas

- ♦ Memo:
 - ♦ Mecanismo para evitar duplicidad.
 - ♦ Almacena todos los árboles y sub-árboles generados.
 - ♦ Almacena todos los planes generados.
 - ♦ Cuando el Optimizador termina, recorre desde el nodo raíz del Memo y elige la mejor alternativa de cada nodo.

El Optimizador de consultas

- ♦ Arquitectura:
 - ♦ Pre-Optimización
 - ♦ Simplificación.
 - ♦ Plan trivial
 - ♦ Exploración
 - ♦ Convertir en plan ejecutable

El Optimizador de consultas

- ♦ Pre-Optimización
 - ♦ Se expande toda la información de las vistas.
 - ♦ Agrupar UNION, INTERSECT y EXCEPT.

El Optimizador de consultas

- ♦ Simplificación.
- ♦ Reorganizar la consulta:
 - ♦ Reordenar JOIN según cardinalidad
 - ♦ Eliminación de contradicciones.
 - ♦ Reorganizar el parámetros del SELECT para que coincidan con las columnas de la tabla.

El Optimizador de consultas

- ♦ Plan trivial / autoparametrización
 - ♦ Planes pregenerados que se suponen dan los mejores resultados.
 - ♦ El Optimizador intentará llegar siempre a un plan trivial.
 - ♦ Un plan trivial equivale a coste 0. Siempre será la mejor opción.

El Optimizador de consultas

- ♦ Plan trivial / autoparametrización
 - ♦ Nunca habrá plan trivial si la consulta presenta:
 - ♦ JOIN
 - ♦ Xpath
 - ♦ Subconsultas
 - ♦ Cursor
 - ♦ Tablas con índices filtrados
- ♦ Se puede intentar forzar.

El Optimizador de consultas

- ♦ Exploración
 - ♦ Uso del Memo
- ♦ Convertir en plan ejecutable

El Optimizador de consultas

- ♦ Optimización de actualizaciones
 - ♦ Protección Halloween
 - ♦ Split/Sort/Collapse
 - ♦ Otras...

El Optimizador de consultas

- ♦ Protección Halloween
 - ♦ Al actualizar, existe la posibilidad de que una fila cambie de posición física.
 - ♦ Puede volver a ser actualizada.
- ♦ Todo plan de actualización se realiza sobre un buffer.

El Optimizador de consultas

- ♦ Split/Sort/Collapse
 - ♦ Estas tres funciones intentan evitar problemas de integridad al hacer actualizaciones sucesivas.

Action	Old	New
UPDATE	1	2
UPDATE	2	3
UPDATE	3	4

El Optimizador de consultas

- ♦ Split/Sort/Collapse
 - ♦ Split: Updates \rightarrow Delete + Insert

Action	Value
DELETE	1
INSERT	2
DELETE	2
INSERT	3
DELETE	3
INSERT	4

El Optimizador de consultas

- ♦ Split/Sort/Collapse
 - ♦ Sort: Reordena por valor, con los Deletes delante.

Action	Value
DELETE	1
DELETE	2
INSERT	2
DELETE	3
INSERT	3
INSERT	4

El Optimizador de consultas

- ♦ Split/Sort/Collapse
 - ♦ Collapse: Elimina Delete y Insert que afecten al mismo valor y los convierte en Update.

Action	Value
DELETE	1
UPDATE	2
UPDATE	3
INSERT	4

El Optimizador de consultas

- ♦ Estadísticas, estimación y coste
 - ♦ Sistema basado en costes.
- ♦ Requiere de una actualización periódica de las estadísticas.
- ♦ Núcleo central del Optimizador

El Optimizador de consultas

- ♦ Estadísticas
 - ♦ Se compone de:
 - ♦ Histograma
 - ♦ Información de las cabeceras (nº filas, ...)
 - ♦ Árboles trie (si la variable es String)
 - ♦ Información de densidad
 - ♦ ...
 - ♦ Guarda la información de todos los datos que soportan comparaciones

El Optimizador de consultas

- Estimación de cardinalidad.
 - Cada operador estima cuantas filas va a procesar (selectividad).
 - Analizando de abajo hacia arriba en el árbol permite estimar cuantas filas serán devueltas.

El Optimizador de consultas

- ♦ Estimación de cardinalidad.
 - ♦ No estima en base a la dependencia entre los datos de distintas columnas.
 - ♦ En caso de errores acumulativos, cuanto más abajo ocurra y más profundo sea el árbol más significativo.
 - ♦ Los operadores poco frecuentes o más nuevos no tienen estimaciones tan precisas.

El Optimizador de consultas

- ♦ Coste
 - ♦ Proceso de estimar cuanto tiempo va a requerir.
 - ♦ Estimación de cardinalidad + media/máxima tamaño de columna + otros factores físicos + supuestos.

Herramientas de diagnóstico

- ♦ Herramientas integradas en Win. y Sql S.
- ♦ Identificar cuellos de botella
 - ♦ Asignar más recursos por SW
 - ♦ Configurar o actualizar HW
- ♦ Errores de diseño en nuestra aplicación
 - ♦ Falta de índices, an. de estadísticas, fragmentación, procesos bloqueantes, aprovechamiento de pl. de ejecución...

Herramientas de diagnóstico

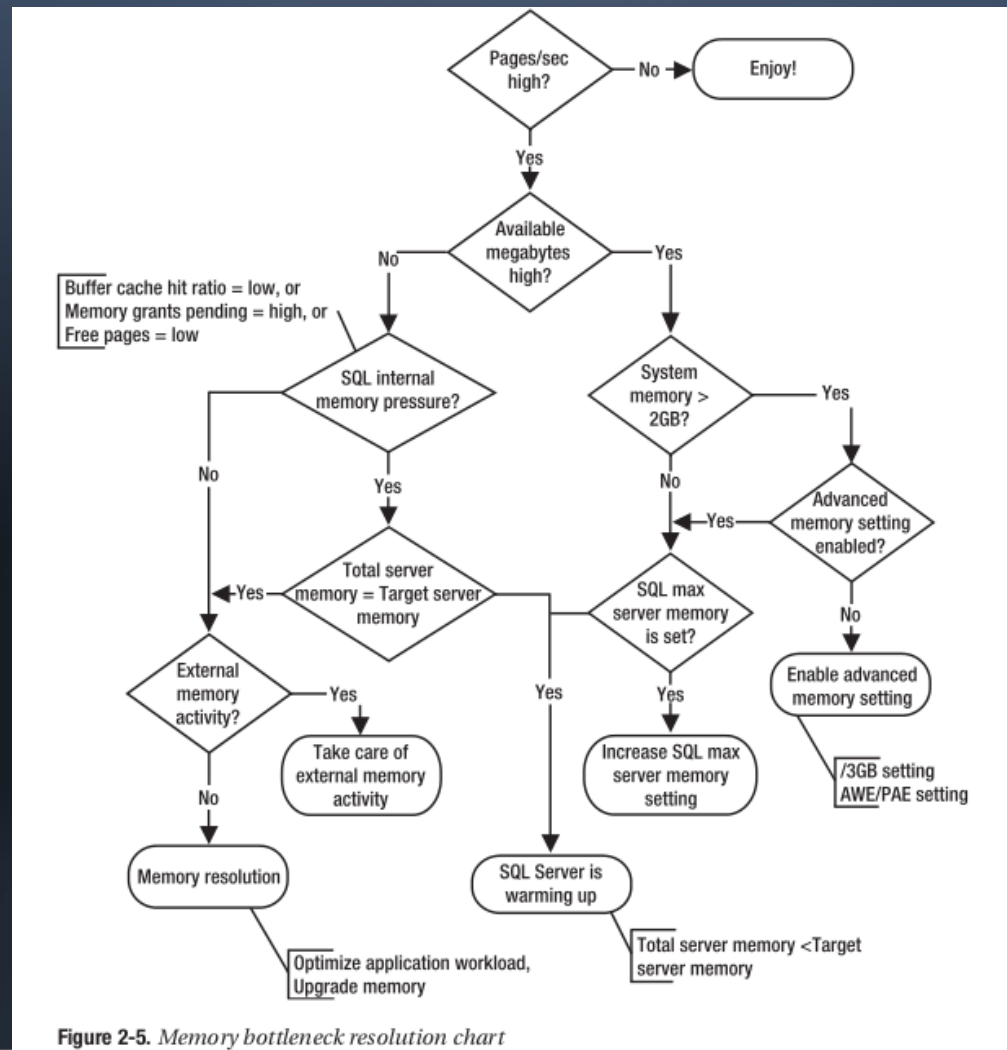


Figure 2-5. Memory bottleneck resolution chart

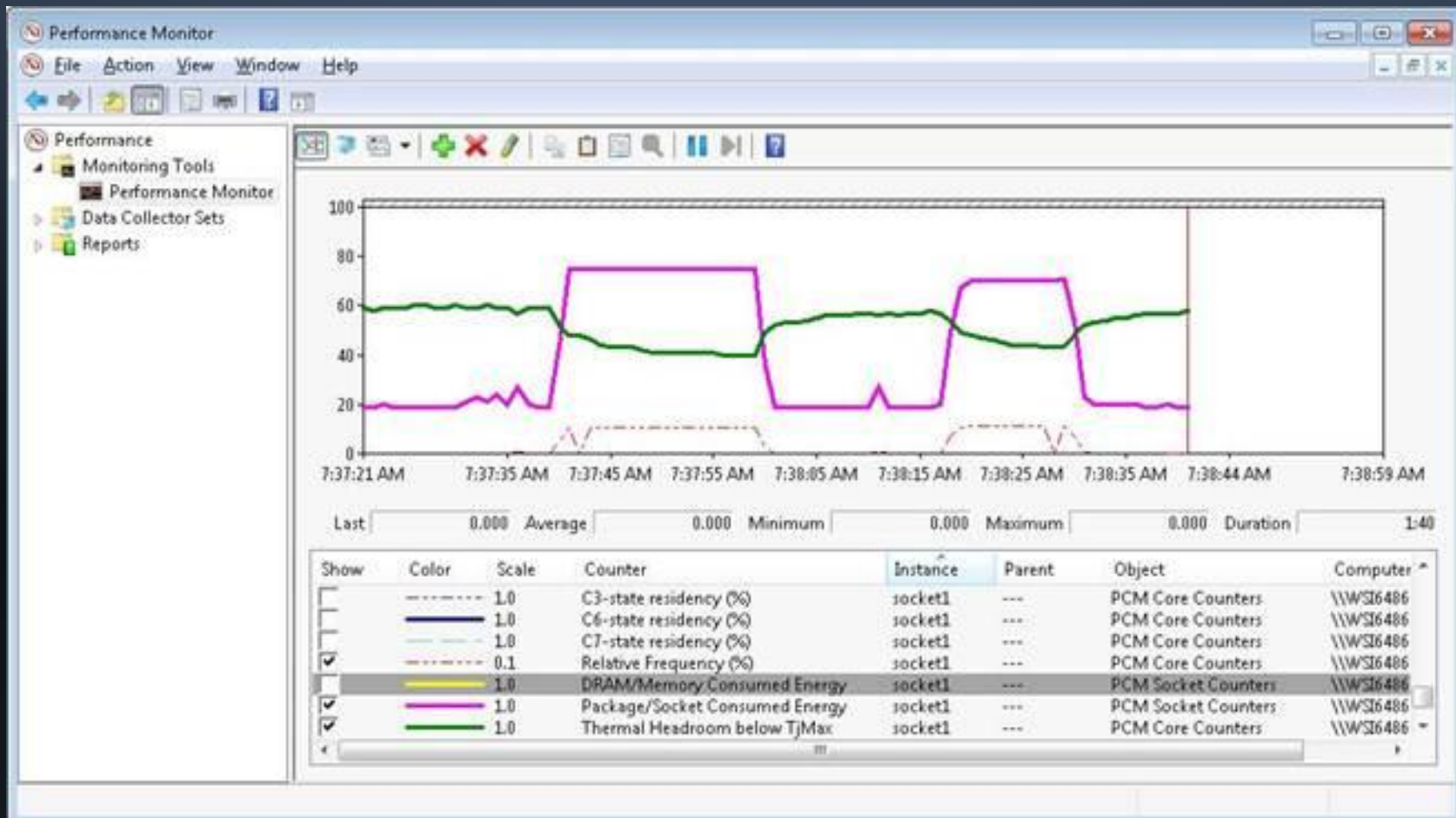
Herramientas de diagnóstico

- System Monitor (perfmon.exe)
- Monitorización del sistema: memoria, disco, procesadores, red, cachés...
- Monitorización de servicios (SQLSERVER, ASP.NET...)

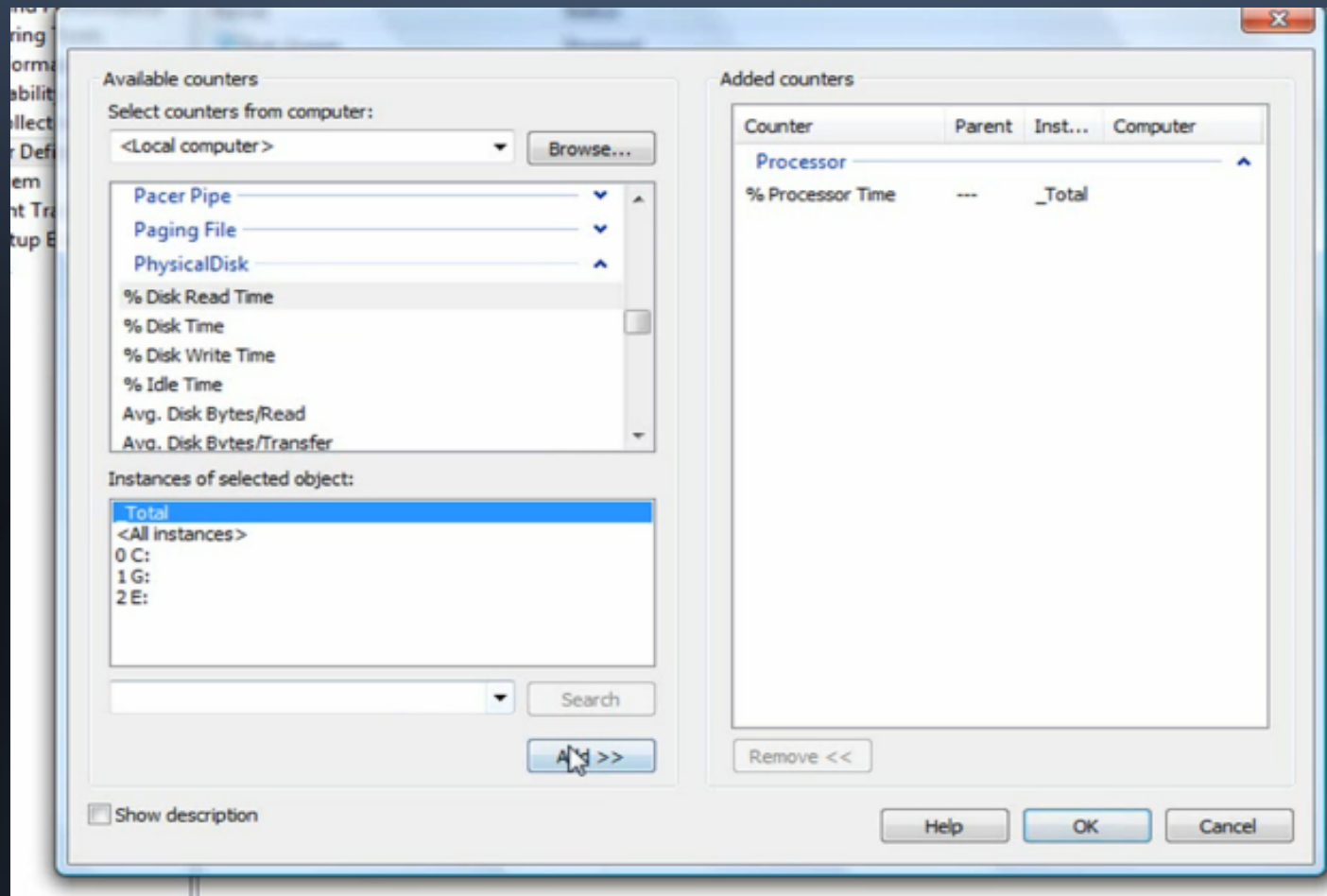
Herramientas de diagnóstico

Objeto	Instancia	Contador/es
SQLSERVER	Access Methods (posible falta de índices)	Free space scans, full scans, table lock escalations, worktables created
SQLSERVER	Latches (sincronismo por integridad)	Total latch wait time
SQLSERVER	Locks (consultas que tardan demasiado + sys.dm_exec_query_stats)	Lock timeouts, lock wait time, number of deadlocks
SQLSERVER	Sql statistics	Batch requests, sql re-compilations
SQLSERVER	General statistics	Processes blocked, user connections

Herramientas de diagnóstico



Herramientas de diagnóstico



Herramientas de diagnóstico

- ♦ El coste de monitorizar puede perjudicar al propio rendimiento del SGBD y falsear las mediciones
- ♦ Data collectors sets (logs)
- ♦ No usar intervalos cortos en el análisis
- ♦ Monitorizar remotamente
- ♦ Almacenar los logs en discos separados
- ♦ Usarlo con tareas programadas

Herramientas de diagnóstico

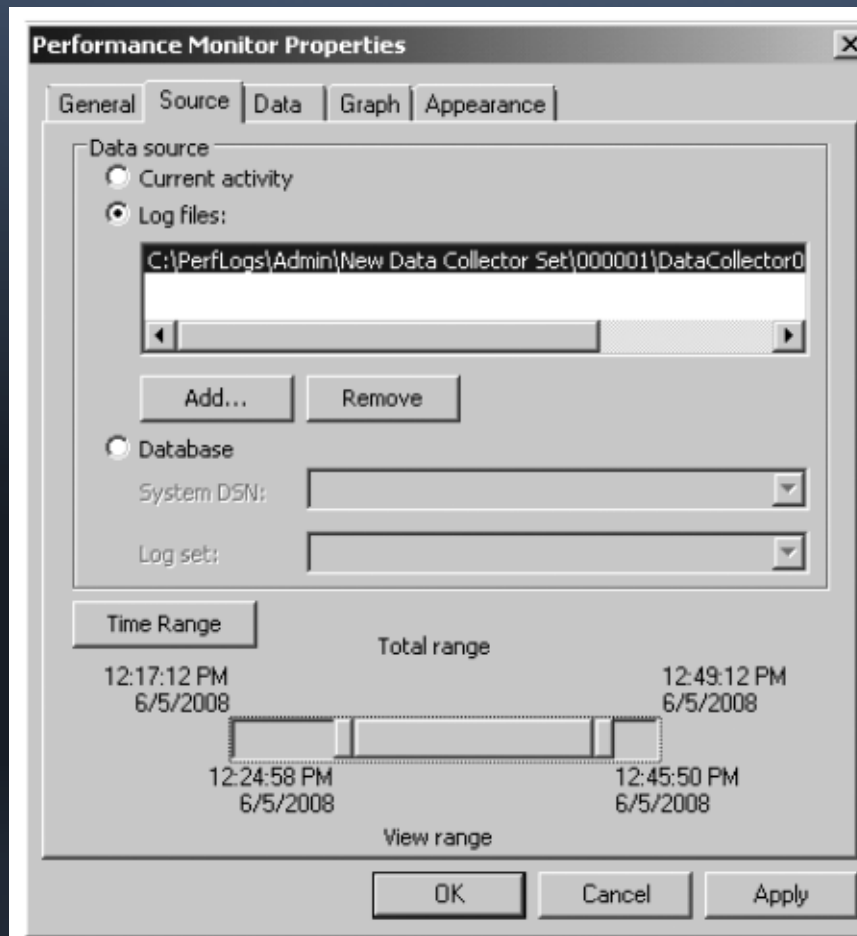


Figure 2-12. *Defining time range for log analysis*

Herramientas de diagnóstico

- ♦ Dynamic Management Views
- ♦ Históricos de rendimiento
- ♦ Se usa en conjunto con lo anterior
- ♦ También disponemos de procedimientos de diagnóstico

Herramientas de diagnóstico

- ♦ `sys.dm_os_performance_counters`
- ♦ `sys.dm_os_waiting_tasks`
- ♦ `sys.dm_os_wait_stats`
- ♦ `sys.dm_exec_requests`
- ♦ `sys.dm_exec_query_stats`
- ♦ `sys.dm_exec_cached_plans` y
`sys.dm_exec_query_plans`
- ♦ `sys.dm_exec_query_memory_grants`

Herramientas de diagnóstico

```
SELECT TOP (10)
    *
FROM    sys.dm_os_wait_stats
ORDER BY wait_time_ms DESC
```

Figure 2-2 displays the output.

	wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wait_time_ms
1	FT_IFTS_SCHEDULER_IDLE_WAIT	448	35901278	600001	1
2	LAZYWRITER_SLEEP	12280	12258720	1460	607
3	SQLTRACE_BUFFER_FLUSH	3064	12255623	4597	6
4	REQUEST_FOR_DEADLOCK_SEARCH	2452	12255005	5003	12255005
5	XE_TIMER_EVENT	410	12241728	30002	12240002
6	SLEEP_TASK	19995	6228098	2967	333
7	BROKER_TO_FLUSH	5979	6126764	1069	28
8	LOGMGR_QUEUE	64	4888389	3643154	2682
9	CHECKPOINT_QUEUE	4	105705	104426	3
10	BROKER_TASK_STOP	6	38912	10000	0

Figure 2-2. *Output from sys.dm_os_wait_stats*

Herramientas de diagnóstico

- Más información:

Query optimizer, query plans and statistics

[<http://technet.microsoft.com/en-us/library/cc966413.aspx>]

Herramientas de diagnóstico

- ♦ Sql Profiler
- ♦ Analiza la carga de trabajo del sistema mediante la captura de eventos
- ♦ Depuración SP
- ♦ Permite reproducir un suceso en el simulador

Herramientas de diagnóstico

The screenshot shows the 'Trace Properties' dialog box with the 'Events Selection' tab selected. The dialog is titled 'Trace Properties' and has a close button (X) in the top right corner. The 'General' tab is also visible. The 'Events Selection' tab contains the following fields and options:

- Trace name:** A text box containing 'Untitled - 3'.
- Trace provider name:** A text box containing 'FRITCHEYGXP\GF2008'.
- Trace provider type:** A text box containing 'Microsoft SQL Server 2008'.
- version:** A text box containing '10.0.1600'.
- Use the template:** A dropdown menu showing 'Standard (default)'.
- Save to file:** A checkbox that is unchecked. Below it is a text box for the file path, a 'Browse' button (folder icon), and a 'Set maximum file size (MB):' text box containing '5'. There are also two checkboxes: 'Enable file rollover' (checked) and 'Server processes trace data' (unchecked).
- Save to table:** A checkbox that is unchecked. Below it is a text box for the table name, a 'Browse' button (table icon), and a 'Set maximum rows (in thousands):' text box containing '1'.
- Enable trace stop time:** A checkbox that is unchecked. Below it are two date/time pickers: the first shows '8/24/2008' and the second shows '8:36:22 AM'.

At the bottom right of the dialog are three buttons: 'Run', 'Cancel', and 'Help'.

Herramientas de diagnóstico

Trace Properties

General Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcess
Security Audit									
<input checked="" type="checkbox"/> Audit Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Audit Logout		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sessions									
<input checked="" type="checkbox"/> ExistingConnection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
Stored Procedures									
<input checked="" type="checkbox"/> RPC:Completed	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL									
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>

Audit Login
Collects all new connection events since the trace was started, such as when a client requests a connection to a server running an instance of SQL Server.

ApplicationName (1 filter(s) applied)
Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

☐ Show all events
☐ Show all columns

Column Filters...
Organize Columns...

Run Cancel Help

Herramientas de diagnóstico

SQL Server Profiler

File Edit View Replay Tools Window Help

Untitled - 2 (FRITCHEYGPVF2008)

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime	EndTime
Audit Logout		Report Server	fritcheygr	corp\...	0	32	0	11000	1760	52	2008-08-31 10:43:57...	2008-08-31
RPC:Completed	exec sp_reset_connection	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:08...	2008-08-31
Audit Login	-- network protocol: LPC set quote...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:08...	
RPC:Completed	declare @p1 nvarchar(64) set @p1=N...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:08...	2008-08-31
RPC:Completed	declare @p1 nvarchar(64) set @p1=N...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:08...	2008-08-31
SQL:BatchStarting	...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:08...	
SQL:BatchCompleted	...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:08...	2008-08-31
SQL:BatchStarting	...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:08...	
SQL:BatchCompleted	...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:08...	2008-08-31
Audit Logout		Report Server	fritcheygr	corp\...	0	32	0	11000	1760	52	2008-08-31 10:44:08...	2008-08-31
RPC:Completed	exec sp_reset_connection	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:19...	2008-08-31
Audit Login	-- network protocol: LPC set quote...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:19...	
RPC:Completed	declare @p1 nvarchar(64) set @p1=N...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:19...	2008-08-31
RPC:Completed	declare @p1 nvarchar(64) set @p1=N...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:19...	2008-08-31
SQL:BatchStarting	...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:19...	
SQL:BatchCompleted	...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:19...	2008-08-31
SQL:BatchStarting	...	Report Server	fritcheygr	corp\...					1760	52	2008-08-31 10:44:19...	
SQL:BatchCompleted	...	Report Server	fritcheygr	corp\...	0	0	0	0	1760	52	2008-08-31 10:44:19...	2008-08-31

```
declare @@BatchID uniqueidentifier
set @@BatchID = newid()

UPDATE [Notifications] WITH (TABLOCK)
SET [BatchID] = @@BatchID,
    [ProcessStart] = GETUTCDATE(),
    [ProcessHeartbeat] = GETUTCDATE()
FROM (
    SELECT TOP 2 [NotificationID] FROM [Notifications] WITH (TABLOCK) WHERE ProcessStart is NULL and
    (ProcessAfter is NULL or ProcessAfter < GETUTCDATE()) ORDER BY [NotificationEntered]
) AS t1
WHERE [Notifications].[NotificationID] = t1.[NotificationID]

select top 2
```

Trace is running.

Ln 30, Col 1 Rows: 30

Herramientas de diagnóstico

SQL Server Profiler - [Untitled - 1 (NL01-DF166BI)]

File Edit View Replay Tools Window Help

EventClass	ApplicationName	BinaryData	ClientProcessID	DatabaseID	DatabaseName	EventSequence	GroupID	HostName	IntegerData	IsSystem	LineNumber
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53755	2	NL01-...	5		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53756	2	NL01-...	1		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53758	2	NL01-...	368		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53759	2	NL01-...	1		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53771	2	NL01-...	8862		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53772	2	NL01-...	42		
Showplan Statistics Profile	Microsoft SQ...	0X14000...	4552	26	BPrompt_P...	53773	2	NL01-...	1		

Trace Pause

Rows	Executes	StmtText
5	1	Sequence
0	1	--Table-valued function(OBJECT:([BPrompt_Partition].[dbo].[fn_UserCampaignLoginData] AS [CampaignLogin]))
0	1	--Table-valued function(OBJECT:([BPrompt_Partition].[dbo].[fn_UserCampaignBreakData]))
5	1	--Sort(DISTINCT ORDER BY:([AgentShift].[RosterDate] ASC, [CampaignLogin].[CampaignName] ASC, [CampaignLogin].[TeamName] ASC, [AgentShift].[User
5	1	--Compute Scalar(DEFINE:([Expr1091]=[Expr1132], [Expr1092]=[Expr1133], [Expr1093]=[BPrompt_Partition].[dbo].[Ufn_Convert]([Expr1134]), [Ex
5	1	--Sort(DISTINCT ORDER BY:([AgentShift].[RosterDate] ASC, [AgentShift].[UserName] ASC, [CampaignLogin].[UserID] ASC, [CampaignLogin].[
5	1	--Hash Match(Left Outer Join, HASH:([CampaignLogin].[CampaignID], [CampaignLogin].[UserID], [AgentShift].[RosterDate])=([Expr108
0	0	--Compute Scalar(DEFINE:([Expr1132]=dateadd(hour,2),[tempdb].[dbo].[#AgentShiftTable].[ShiftStartDateTime] as [AgentShift]
0	0	--Compute Scalar(DEFINE:([Expr1007]=CONVERT_IMPLICIT(int,[Expr1143],0))
5	1	--Stream Aggregate(GROUP BY:([AgentShift].[RosterDate], [AgentShift].[UserName], [CampaignLogin].[UserID], [Campa
11	1	--Sort(ORDER BY:([AgentShift].[RosterDate] ASC, [AgentShift].[UserName] ASC, [CampaignLogin].[UserID], [Campa
11	1	--Hash Match(Inner Join, HASH:([CampaignLogin].[UserID])=([AgentShift].[UserID]), RESIDUAL:([BPrompt_Pa
0	0	--Compute Scalar(DEFINE:([Expr1135]=isnull([BPrompt_Partition].[dbo].[fn_UserCampaignLoginData].[L
11	1	--Table Scan(OBJECT:([BPrompt_Partition].[dbo].[fn_UserCampaignLoginData] AS [CampaignLogin])
3838	1	--Table Scan(OBJECT:([tempdb].[dbo].[#AgentShiftTable] AS [AgentShift]))
0	0	--Compute Scalar(DEFINE:([Expr1081]=CASE WHEN [Expr1022] IS NOT NULL THEN [Expr1022] ELSE [Expr1071] END, [Expr1082]=CASE W
7676	1	--Hash Match(Full Outer Join, HASH:([AgentShift].[UserID], [AgentShift].[RosterDate], [BPrompt_Partition].[dbo].[fn_Us
0	0	--Compute Scalar(DEFINE:([Expr1022]=[tempdb].[dbo].[#AgentShiftTable].[UserID] as [AgentShift].[UserID], [Expr102
3838	1	--Stream Aggregate(GROUP BY:([AgentShift].[UserID], [AgentShift].[RosterDate], [AgentShift].[UserName], [Age
3838	1	--Sort(ORDER BY:([AgentShift].[UserID] ASC, [AgentShift].[RosterDate] ASC, [AgentShift].[UserName] ASC,

Ready.

Rows: 65

Connections: 1

Herramientas de diagnóstico

- Utilización de la CPU de una consulta o actividad involucrada
- Memoria utilizada por la actividad
- Lecturas y escrituras por la actividad
- Cuanto tardó la actividad
- Con qué frecuencia es realizada
- Los errores que fueron lanzados

Herramientas de diagnóstico

- El resultado de una actividad se obtiene tras completarse algún evento asociado
- Finalización SP o script TSQL, realización de auditoría, lanzamiento de excepciones, recompilaciones, transacciones...
- Los resultados se pueden almacenar después en un fichero o en una tabla

Herramientas de diagnóstico

- ♦ En caso de no disponer de la herramienta: `sys.dm_exec_query_stats`
- ♦ Obtenemos un enlace a fichero XML con el plan de ejecución*
- ♦ Este fichero XML es cargado en la aplicación y nos muestra el plan gráficamente

Herramientas de diagnóstico

- ♦ *Un plan de ejecución es la estrategia de procesamiento usada por el optimizador de consultas de Sql Server al ejecutar una consulta, basándose en estadísticas de uso, correlación de datos y configuración de las tablas
- ♦ Su estudio nos permite aprender a optimizar nuestras consultas

Herramientas de diagnóstico

- ♦ Mediante opciones SET para consultas ad-hoc
- ♦ SHOWPLAN_XML (y STATISTICS XML, STATISTICS IO...)

Herramientas de diagnóstico

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the AdventureWorksLT2008 database selected. The central pane contains a T-SQL query titled "set_showplan.s_ritcheyg (55)*". The query is as follows:

```
USE AdventureworksLT2008
GO
SET SHOWPLAN_XML ON
GO
SELECT soh.AccountNumber,
       sod.LineTotal,
       sod.OrderQty,
       sod.UnitPrice,
       p.Name
FROM   SalesLT.SalesOrderHeader soh
JOIN   SalesLT.SalesOrderDetail sod
      ON soh.SalesOrderID = sod.SalesOrderID
JOIN   SalesLT.Product p
      ON sod.ProductID = p.ProductID
WHERE  sod.LineTotal > 1000
GO
SET SHOWPLAN_XML OFF
GO
```

The right pane shows the Properties window for the current connection parameters.

Below the query, the Execution Plan tab is active, displaying a graphical representation of the query's execution strategy. The plan includes the following operators and their estimated costs:

- Select**: Cost: 0%
- Hash Match (Inner Join)**: Cost: 44%
- Compute Scalar**: Cost: 0%
- Nested Loops (Inner Join)**: Cost: 2%
- Clustered Index Scan (Clustered) [SalesOrderHeader].[PK_SalesOrderHeader]**: Cost: 6%
- Compute Scalar**: Cost: 0%
- Clustered Index : [SalesOrderDetail].[PK_SalesOrderDetail]**: Cost: 0%
- Index Scan (NonClustered) [Product].[AK_Product_Name]**: Cost: 0%

The bottom pane shows the XML output of the execution plan, which includes details about the physical operations, logical operations, and the actual execution statistics.

Herramientas de diagnóstico

- ♦ El grosor de las líneas que interconectan los nodos indican un alto número de filas transferidas
- ♦ Cada nodo lleva asociado el porcentaje del coste total de la operación
- ♦ Nos indica el tipo de agrupación utilizado, el índice escogido, etc...

Herramientas de diagnóstico

- La diferencia entre SHOWPLAN y STATISTICS es que showplan no realiza la consulta (solo te muestra la información acerca del plan escogido para la consulta) y statistics sí.
- Por eso statistics es capaz de decirte el tiempo de parseo, compilación y ejecución y el número de lecturas escrituras en los discos de cada una de las tablas.

Herramientas de diagnóstico

- Statistics también te ofrece capturar el coste de la consulta desde la perspectiva del cliente para comprobar el tiempo que tarda en transmitirse los datos a través de la red

Índices en SQL Server 2008

- Los índices evitan que el motor de consultas no tenga que examinar una tabla entera, pero por contra se requiere de más espacio en disco y de una leve penalización en las operaciones no select por reestructuración y mantenimiento de ellas

Índices en SQL Server 2008

- Recomendaciones en la creación de índices

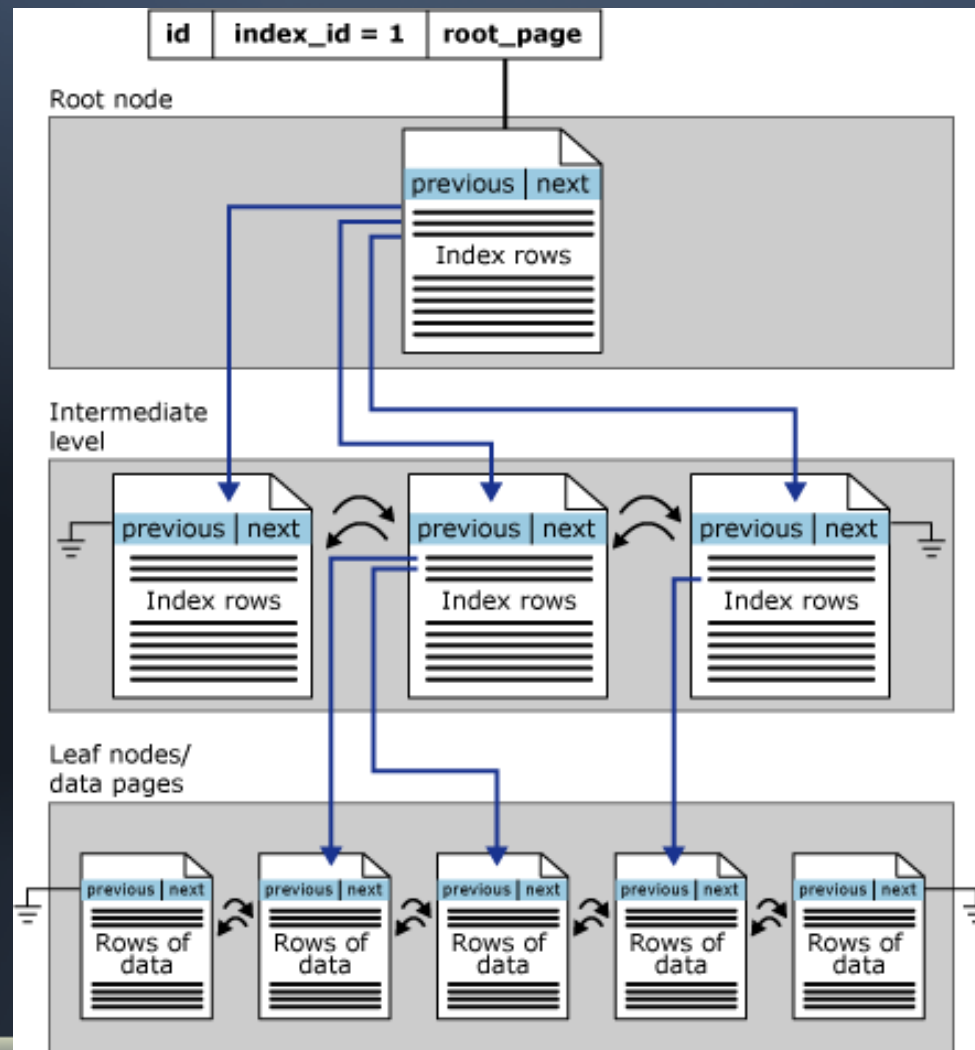
Índices en SQL Server 2008

- Considerar el tipo de índice (clustered (agrupado) vs. nonclustered (no agrupado))
- Ambos índices tienen una estructura de árbol B+

Índices en SQL Server 2008

- ♦ En un índice clustered los nodos hoja del árbol corresponden con los datos de la tabla, en el mismo orden en el que los datos se almacenan.
- ♦ Por lo tanto este índice no ocupa memoria
- ♦ La página de datos del tamaño especificado por el parámetro "fill factor page index"

Índices en SQL Server 2008



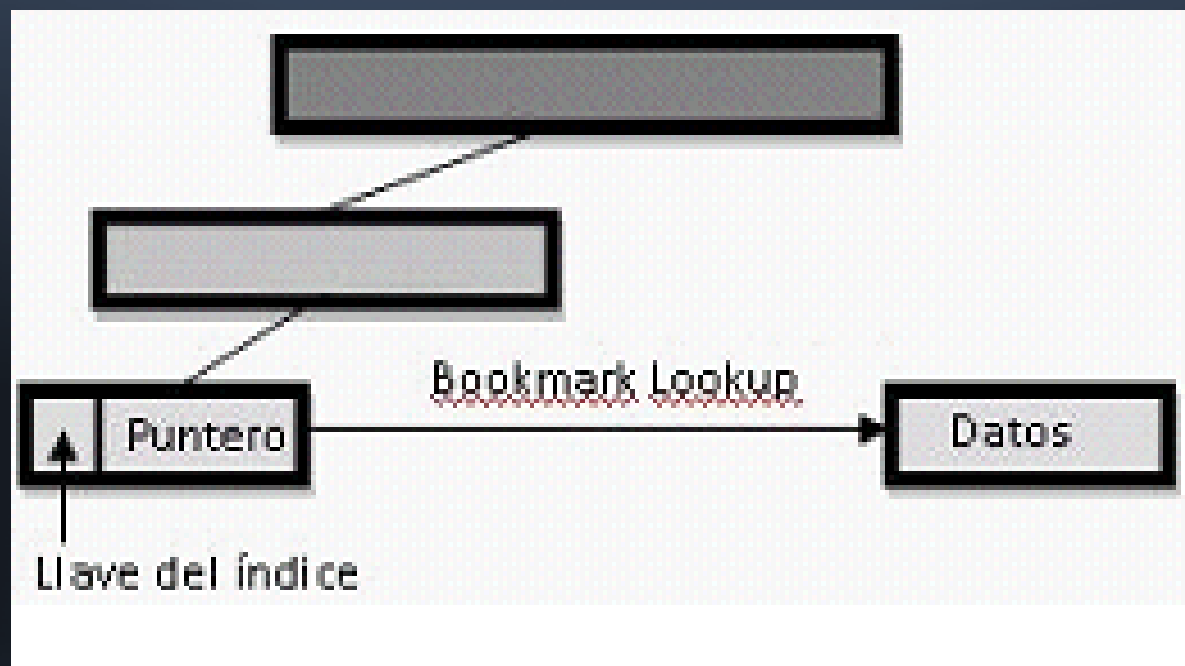
Índices en SQL Server 2008

- Las columnas del índice deben de estar ordenadas (primary key) aunque puede haber datos repetidos
- Si no, no sirve de nada (por acceso a páginas distintas por cada fila)
- Por defecto, SQL Server crea un índice clustered con la clave primaria cada vez que creamos una tabla

Índices en SQL Server 2008

- La estructura de un índice non-clustered o no agrupado, también llamada tablas heap, no tiene que estar ordenada de ninguna forma en particular.
- Sus nodos hoja son punteros a los datos de la tabla

Índices en SQL Server 2008



Índices en SQL Server 2008

- La clave del índice está repartida en las páginas del índice, de modo tal que la búsqueda se haga leyendo la menor cantidad posible de datos

Índices en SQL Server 2008

- ♦ Por lo tanto, el uso de este tipo de índices hace necesario hacer lecturas adicionales para traer el registro a través de los punteros (bookmark lookup o salto a la página de datos).
- ♦ Estos índices si que ocupan espacio y deberían de estar en un disco aparte

Índices en SQL Server 2008

- ♦ Solución: incluir dentro del nodo hoja del índice campos que no son parte de la clave (key lookups)
- ♦ Key lookups Implica más espacio en disco

Índices en SQL Server 2008

```
CREATE NONCLUSTERED INDEX  
    AK_Employee_NationalIDNumber  
    ON Purchasing.ProductVendor (  
        BusinessEntityID  
    ) INCLUDE (JobTitle, HireDate)  
WITH drop_existing
```

Índices en SQL Server 2008

- ◆ Clustered vs Nonclustered
 - ◆ + Agrupado
 - ◆ Optimo para lectura por rangos o consultas con ORDER BY o GROUP BY
 - ◆ Optimo para lecturas de muchas filas
 - ◆ SQL Server descarta los índices no agrupados cuando tiene que devolver una gran cantidad de filas
 - ◆ Un índice agrupado como máximo
 - ◆ Lo aplicaremos para aquellas columnas que suelen ser muy consultadas y no contengan muchos duplicados

Índices en SQL Server 2008

- ◆ Clustered vs Nonclustered
 - ◆ + Nonclustered
 - ◆ Lectura de pocas filas con cláusulas WHERE que devuelven pocos registros
 - ◆ Múltiples índices no agrupados
 - ◆ Es posible tener más de un índice por tabla

Índices en SQL Server 2008

- ◆ Clustered vs Nonclustered
 - ◆ - Nonclustered
 - ◆ Si hay actualización frecuente
 - ◆ Una actualización de datos implica actualizar todas las referencias a los datos de la tabla
 - ◆ No recomendable usar muchos índices de este tipo sobre una tabla por esta razón

Índices en SQL Server 2008

- Recomendaciones en la creación de índices (bis)

Índices en SQL Server 2008

- Examinar la clausula WHERE y las condiciones del JOIN para escoger el mejor índice
- Existe una herramienta integrada junto con Sql Server Profiler llamada Database Engine Tuning Advisor (dta.exe), que evalúa tu esquema de datos junto con una batería de consultas o un log de carga de trabajo del Sql Profiler que le suministres

Índices en SQL Server 2008

- Realiza recomendaciones sobre cuando es necesario eliminar índices o crear nuevos y en base a qué columnas
- Es necesario saber interpretar los resultados porque no siempre resultan fiables

Índices en SQL Server 2008

- Usar índices de pocas columnas y columnas de tipos de no excesivo tamaño
- Cuanto más grande sea la clave por la que se indiza una página de datos mayor será el tiempo de acceso

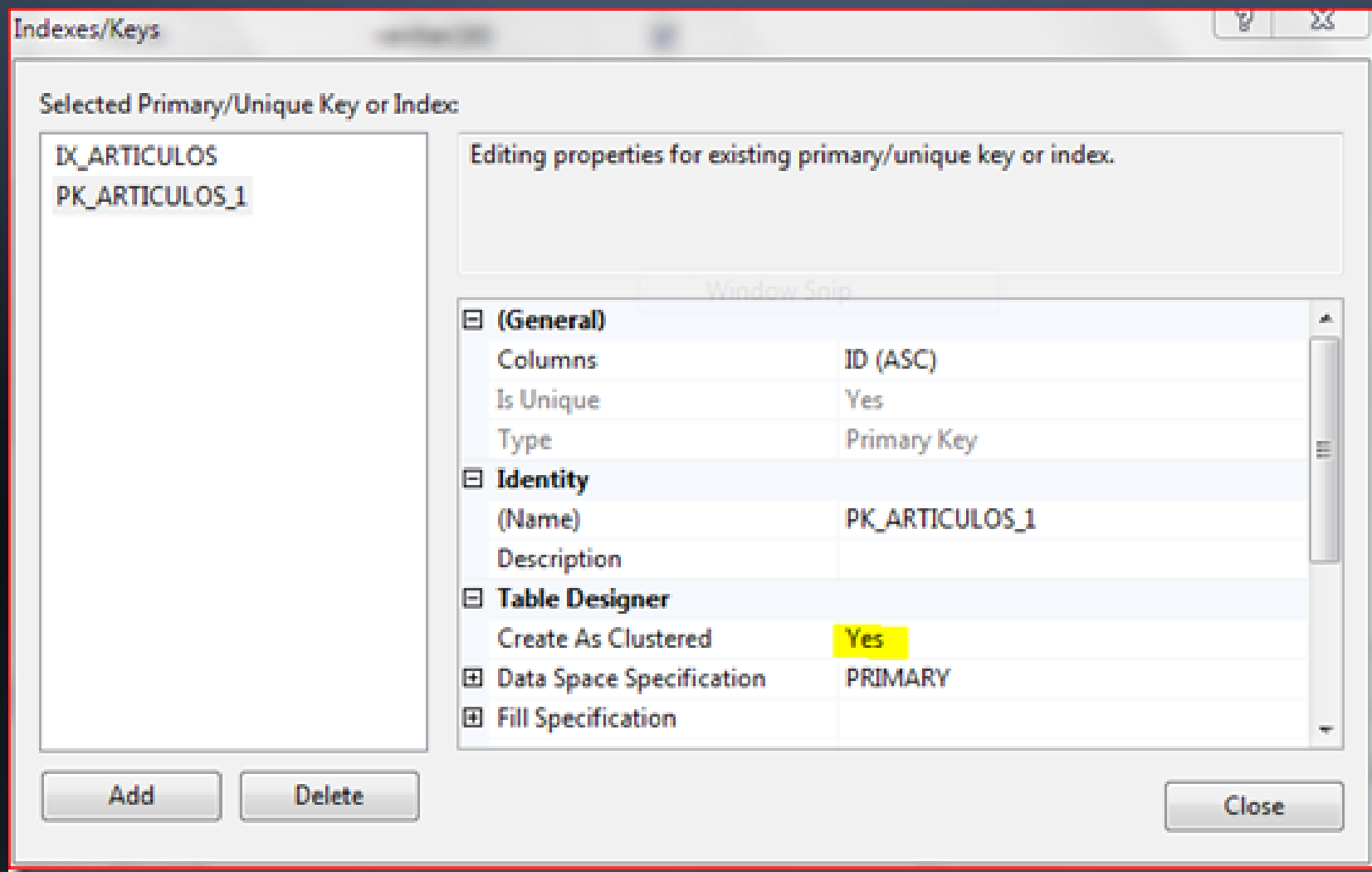
Índices en SQL Server 2008

- Considerar el orden de los datos de las columnas según sus datos
- Menor será el tiempo de construcción del índice

Índices en SQL Server 2008

- ¿El conjunto de columnas del índice debe de ser *unique*?
- Depende del tipo de consultas que vayamos a realizar
- Saber escoger entre “clustered y nonclustered” para cada caso.
- [Ejemplo]

Índices en SQL Server 2008



Índices en SQL Server 2008

```
create table movimientos (id bigint identity PRIMARY KEY,  
                           articulo varchar(50),  
                           fecha datetime,  
                           cantidad int  
                           )
```

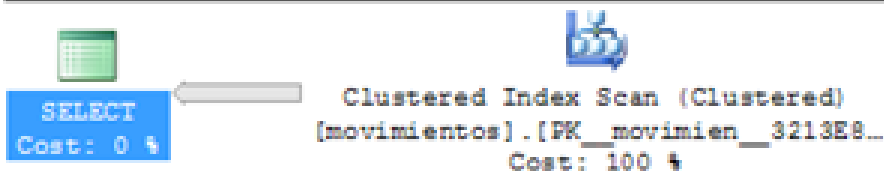
	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	PRIMARY KEY (clustered)	PK_movimien__3213E83F1B0907CE	(n/a)	(n/a)	(n/a)	(n/a)	id

Índices en SQL Server 2008

Query 1: Query cost (relative to the batch): 94%

```
select articulo from movimientos where fecha >= '20091101' and fecha < '20100101'
```

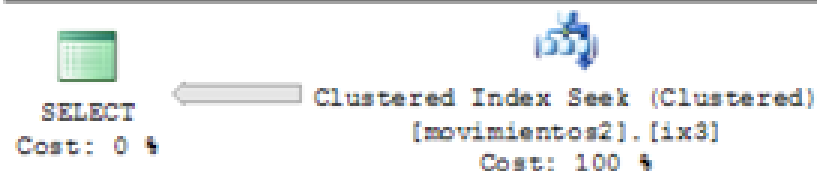
Missing Index (Impact 71.109): CREATE NONCLUSTERED INDEX [<Name of Missing Index, s



Query 2: Query cost (relative to the batch): 6%

```
select articulo from movimientos2 where fecha >= '20091101' and fecha < '20100101'
```

Missing Index (Impact 71.109): CREATE NONCLUSTERED INDEX [<Name of Missing Index, s



Índices en SQL Server 2008

	index_name	index_description	index_keys
1	ix2	nonclustered located on PRIMARY	fecha
2	PK__movimien__3213E83F20C1E124	clustered, unique, primary key located on PRIMARY	id

	index_name	index_description	index_keys
1	ix3	clustered located on PRIMARY	fecha
2	ix4	nonclustered located on PRIMARY	id

Índices en SQL Server 2008

```
select articulo  
from movimientos  
where fecha >= '20091101'  
and fecha < '20100101'
```

```
select articulo  
from movimientos2  
where fecha >= '20091101'  
and fecha < '20100101'
```


Índices en SQL Server 2008

- ◆ Índices especiales
 - ◆ Full-text, para trabajar con índices binarios
 - ◆ Spatial, para índices de tipo geográficos o geométricos, es posible crear hasta 249 índices por columna
 - ◆ XML, para trabajar con documentos xml con XQuery

Índices en SQL Server 2008

- ♦ Índices especiales
 - ♦ Índices filtrados
 - ♦ Se utilizan cuando la cláusula WHERE definida en la creación del índice coincide con la condición WHERE de la consulta realizada
 - ♦ Ocupan menos espacio en disco que los índices completos y son más sencillos de mantener

Índices en SQL Server 2008

- ◆ Estadísticas
 - ◆ Para qué sirven
 - ◆ Para la optimización del plan de ejecución de consultas.
 - ◆ Utiliza heurísticas con dichas estadísticas junto con la distribución de datos en nuestra BD y las últimas consultas realizadas.
 - ◆ Con ello el optimizador entre otras cosas, intenta escoger el índice más apropiado en cada momento

Índices en SQL Server 2008

- ♦ El estado de nuestra BD es cambiante
- ♦ Las estadísticas se pueden quedar desactualizadas
- ♦ `ALTER DATABASE YourDataBase SET AUTO_UPDATE_STATISTICS ON` (por defecto)
- ♦ `DBCC SHOW_STATISTICS (table_name, index_name)`

Índices en SQL Server 2008

- ♦ Sin embargo, actualizar las estadísticas es un trabajo pesado.
- ♦ Si no decimos nada, es SQL Server quien las actualiza cuando cree conveniente
- ♦ Podemos hacerlo manualmente con `sp_updatestats`
- ♦ Especificar este procedimiento como una tarea programada

Índices en SQL Server 2008

- Fragmentación en índices
- Provocado por el continuo reordenamiento, creación y división de páginas de índices y de filas dentro de dichas páginas al realizar modificaciones en los datos

Índices en SQL Server 2008

- Se provoca fragmentación externa cuando producto de la reordenación de páginas, datos que deberían estar contiguos en una misma extensión (unidad física de almacenamiento en disco) lo están en extensiones diferentes (cada extensión tiene un tamaño de páginas limitado) provocando lentitud en el acceso y desaprovechamiento de espacio dentro de las páginas (fragmentación interna)

Índices en SQL Server 2008

- ♦ `sys.dm_db_index_physical_stats`
- ♦ Información acerca del tamaño de los índices y su fragmentación
- ♦ Desfragmentación
 - ♦ ¿Cómo?

Índices en SQL Server 2008

Characteristics/Issues	Drop and Create Index	Create Index with DROP_EXISTING	ALTER INDEX REBUILD	ALTER INDEX REORGANIZE
Rebuild nonclustered indexes on clustered index fragmentation	Twice	No	No	No
Missing indexes	Yes	No	No	No
Defragment index with constraints	Highly complex	Moderately complex	Easy	Easy
Defragment multiple indexes together	No	No	Yes	Yes
Concurrency with others	Low	Low	Medium, depending on concurrent user activity	High
Intermediate cancellation	Dangerous with no transaction	Progress lost	Progress lost	Progress preserved
Degree of defragmentation	High	High	High	Moderate to low
Apply new fill factor	Yes	Yes	Yes	No
Statistics are updated	Yes	Yes	Yes	No

Índices en SQL Server 2008

- ♦ Por último
- ♦ Aspectos relacionados con procedimientos almacenados
- ♦ Es el modo más eficiente de obtener datos

Índices en SQL Server 2008

- ♦ Cuando un procedimiento se compila (se asocia un plan de ejecución a dicho proceso) por primera vez o se vuelve a compilar (si estado o estructura de la base de datos ha cambiado lo suficiente)
- ♦ El plan de consulta asociado en el proceso de compilación del procedimiento se optimiza para el estado actual de la base de datos.

Índices en SQL Server 2008

- ♦ Por lo tanto nos ahorramos la mayoría de veces el proceso de obtener un nuevo plan en cada ejecución
- ♦ Sin embargo no siempre puede ser útil reutilizar los planes de ejecución cacheadas
- ♦ Se puede controlar el proceso de recompilación en aquellos casos particulares

Índices en SQL Server 2008

- ♦ Sys.dm_exec_cached_plans
- ♦ Y además
 - ♦ Reducen el tráfico de datos en la red
 - ♦ Aislan el acceso a los datos en tu aplicación
 - ♦ Ofrecen políticas de seguridad contra intrusos y un único punto de administración