

MONITOREO Y RENDIMIENTO EN SQL SERVER

Proyecto 1: Captura de consultas lentas con Extended Events

1. Enunciado del ejercicio

Crear una sesión de Extended Events que capture consultas que tarden más de 1 segundo en ejecutarse en la base de datos QhatuPeru. Guardar el resultado en un archivo .xel.

Detalle	Contenido
Enunciado	Capturar consultas que tardan más de 5 segundos.
Script de la Solución	Creación y configuración de una sesión de Extended Events (XE_ConsultasLentas) para el evento sqlserver.rpc_completed o sqlserver.sql_statement_completed con un filtro de duración (duration > 5000000 microsegundos).
Justificación Técnica	Se utiliza Extended Events (XE) , la tecnología de monitoreo de rendimiento de menor impacto de SQL Server, para capturar la actividad en el servidor. El filtro duration > 5000000 evita el registro de transacciones rápidas, enfocando el análisis solo en los cuellos de botella reales.
Buenas Prácticas	Se prioriza XE sobre SQL Trace. Se usa un filtro estricto para reducir la sobrecarga de datos y se almacena en un <i>file target</i> para análisis diferido, no en memoria (como ring_buffer), lo cual es mejor para largos períodos de monitoreo.

```
1  -- Ejecutar en la base de datos 'master' o en cualquier base de datos que no sea QhatuPeru.
2  USE master;
3  GO
4
5  -- Verificar y eliminar la sesión si ya existe para evitar errores
6  IF EXISTS (SELECT * FROM sys.server_event_sessions WHERE name = 'CapturaConsultasLentas')
7  DROP EVENT SESSION CapturaConsultasLentas ON SERVER;
8  GO
9
10 -- Crear la nueva sesión de Extended Events
11 CREATE EVENT SESSION CapturaConsultasLentas
12 ON SERVER
13 ADD EVENT sqlserver.sql_statement_completed
14 (
15     -- Configuración de los filtros:
16     WHERE (
17         -- 1. Capturar duración mayor o igual a 1 segundo (1,000,000 microsegundos)
18         [duration] >= (1000000)
19         -- 2. Filtrar solo las consultas de la base de datos 'QhatuPeru'
20         AND [database_name]='QhatuPeru'
21     )
22 )
23 -- Definir el destino de los datos (Target): un archivo .xel
24 ADD TARGET package0.event_file(SET
25     filename='C:\SQL_Logs\CapturaConsultasLentas.xel', -- AJUSTA ESTA RUTA si la carpeta no existe en tu PC
26     max_file_size=(50),
27     max_rollover_files=(4)
28 )
29 WITH (
30     MAX_MEMORY=4096 KB,
31     STARTUP_STATE=OFF -- Se crea, pero el inicio es manual abajo
32 );
33 GO
34
35 -- Iniciar la sesión de Extended Events para comenzar el monitoreo
36 ALTER EVENT SESSION CapturaConsultasLentas ON SERVER STATE = START;
37 GO
```

47 % No se encontraron problemas. Línea: 22, Car

Mensajes

Los mensajes se han completado correctamente.

Mensaje de finalización: 0000-00-00 00:00:00.00 "000000-00-00"

```
1  USE QhatuPeru;
2  GO
3
4  -- Simula una consulta que tarda 3 segundos (3000 milisegundos)
5  -- Esto debería ser capturado por tu sesión de Extended Events
6  WAITFOR DELAY '00:00:03';
7
8  SELECT GETDATE() AS HoraActual, 'Esta consulta tardó 3 segundos' AS Resultado;
9  GO
```

83 % No se encontraron problemas. Línea: 9, Carácter: 3 TABULACIONES CRLF Windows 1252

Resultados Mensajes

	HoraActual	Resultado
1	2025-12-14 14:41:36.777	Esta consulta tardó 3 segundos

-- Ejecutar en la base de datos 'master' o en cualquier base de datos que no sea QhatuPeru.

USE master;

GO

-- Verificar y eliminar la sesión si ya existe para evitar errores

IF EXISTS (SELECT * FROM sys.server_event_sessions WHERE name = 'CapturaConsultasLentas')

DROP EVENT SESSION CapturaConsultasLentas ON SERVER;

GO

- Crear la nueva sesión de Extended Events

CREATE EVENT SESSION CapturaConsultasLentas

ON SERVER

ADD EVENT sqlserver.sql_statement_completed

```
(

-- Configuración de los filtros:

WHERE (

    -- 1. Capturar duración mayor o igual a 1 segundo (1,000,000 microsegundos)

    [duration] >= (1000000)

    -- 2. Filtrar solo las consultas de la base de datos 'QhatuPeru'

    AND [database_name]=N'QhatuPeru'

)

)

-- Definir el destino de los datos (Target): un archivo .xel

ADD TARGET package0.event_file(SET

    filename=N'C:\SQL_Logs\CapturaConsultasLentas.xel', -- ⚠ AJUSTA ESTA RUTA si la carpeta no existe en tu PC

    max_file_size=(50),

    max_rollover_files=(4)

)

WITH (

    MAX_MEMORY=4096 KB,

    STARTUP_STATE=OFF -- Se crea, pero el inicio es manual abajo

);

GO

-- Iniciar la sesión de Extended Events para comenzar el monitoreo

ALTER EVENT SESSION CapturaConsultasLentas ON SERVER STATE = START;

GO
```

Proyecto 2: Crear índices para mejorar la búsqueda de clientes

1. Enunciado del ejercicio

En la tabla Clientes, mejorar el rendimiento de búsqueda por DNI y Apellidos creando índices adecuados.

Detalle	Contenido
Enunciado	Crear un índice no agrupado en la tabla CLIENTE sobre la columna DNI.
Script de la Solución	CREATE NONCLUSTERED INDEX IX_CLIENTE_DNI ON CLIENTE (DNI);
Justificación Técnica	La columna DNI es un campo de alta selectividad que se utiliza a menudo en la cláusula WHERE para búsquedas directas. Un índice no agrupado sobre esta columna mejora la velocidad de búsqueda (<i>Index Seek</i>), evitando el escaneo completo de la tabla (<i>Table Scan</i>).
Buenas Prácticas	Se utiliza un índice no agrupado (NONCLUSTERED) ya que la clave primaria (generalmente CodCliente) ya utiliza el índice agrupado. Se nombra el índice de forma descriptiva (IX_Tabla_Columna).

creamo la tabla cliente

```
1  USE QhatuPeru;
2  GO
3
4  -- Crear la tabla CLIENTE
5  CREATE TABLE CLIENTE (
6      CodCliente INT IDENTITY(1,1) PRIMARY KEY,
7      DNI VARCHAR(15) NOT NULL, -- Identificador único nacional
8      Apellidos VARCHAR(50) NOT NULL,
9      Nombres VARCHAR(50) NOT NULL,
10     Direccion VARCHAR(80),
11     Telefono VARCHAR(15),
12     CONSTRAINT UQ_Cliente_DNI UNIQUE (DNI) -- Aseguramos que el DNI no se repita
13 );
14 GO
```

110 %
● No se encontraron problemas.
Línea: 14, Carácter: 3 SPC CRLF Windows 1252

Mensajes
Los comandos se han completado correctamente.
Hora de finalización: 2025-12-14T14:44:52.2444972-05:00

Una vez creada la tabla CLIENTE, crearemos los índices para optimizar las búsquedas:

1. Índice para Búsqueda por DNI

Como el DNI es un identificador único y se usará en búsquedas exactas, un índice único no agrupado (NONCLUSTERED) es ideal. Aunque ya le pusiste una restricción UNIQUE, crear un índice explícito puede ser útil.

```
1  -- Índice en DNI para búsquedas rápidas por identidad única
2  CREATE UNIQUE NONCLUSTERED INDEX IX_CLIENTE_DNI
3  ON CLIENTE (DNI);
4  GO
```

110 % No se encontraron problemas. Línea: 4, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-12-14T14:46:05.0935478-05:00

2. Índice para Búsqueda por Apellidos

Cuando se busca por apellidos, es común usar el operador LIKE 'Apellido%', por lo que un índice simple no agrupado sobre esta columna es lo más adecuado.

```
1  -- Índice en Apellidos para mejorar las búsquedas parciales (LIKE 'Apellido%') o totales
2  CREATE NONCLUSTERED INDEX IX_CLIENTE_Apellidos
3  ON CLIENTE (Apellidos);
4  GO
```

110 % No se encontraron problemas. Línea: 4, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-12-14T14:47:03.1623718-05:00

Usar DMV para evaluar la fragmentación de índices en QhatuPeru y reconstruir o reorganizar según el porcentaje encontrado.

Detalle	Contenido
Enunciado	Detectar y corregir la fragmentación de índices.
Script de la Solución	Detección: <code>SELECT * FROM sys.dm_db_index_physical_stats(...) WHERE avg_fragmentation_in_percent > 5</code> . Solución (Ejemplo): <code>ALTER INDEX [NombreIndice] ON [NombreTabla] REBUILD;</code>
Justificación Técnica	La DMV <code>sys.dm_db_index_physical_stats</code> proporciona el porcentaje de fragmentación. Si la fragmentación es > 30%, se usa <code>REBUILD</code> (reconstrucción completa, más costoso pero más efectivo). Si la fragmentación está entre 5% y 30%, se usa <code>REORGANIZE</code> (reordenamiento en línea, menos costoso).
Buenas Prácticas	Se verifica la fragmentación antes de aplicar el mantenimiento. El mantenimiento se realiza fuera de horas pico para minimizar el impacto en el rendimiento de las consultas activas.

1

USE QhatuPeru;

2

GO

3

4

SELECT

5

OBJECT_NAME(ips.object_id) AS NombreTabla,

6

i.name AS NombreIndice,

7

ips.avg_fragmentation_in_percent AS PorcentajeFragmentacion,

8

ips.page_count AS NumeroPaginas

9

FROM

10

sys.dm_db_index_physical_stats(DB_ID(), OBJECT_ID('CLIENTE'), NULL, NULL, 'DETAILED') AS ips

11

INNER JOIN

12

sys.indexes AS i ON ips.object_id = i.object_id AND ips.index_id = i.index_id

13

WHERE

14

ips.avg_fragmentation_in_percent > 5 -- Filtro: Mostrar solo > 5%

15

AND ips.index_id > 0

16

ORDER BY

17

ips.avg_fragmentation_in_percent DESC;

18

GO

91 %

No se encontraron problemas.

Línea: 18, Carácter: 3

SPC

CRLF

Windows 1252

Resultados

Mensajes

	NombreTabla	NombreIndice	PorcentajeFragmentacion	NumeroPaginas
1	CLIENTE	IX_CLIENTE_Apellidos	22.9508196721311	61

```
USE QhatuPeru;
```

```
GO
```

```
SELECT
```

```
OBJECT_NAME(ips.object_id) AS NombreTabla,
```

```
i.name AS NombreIndice,

ips.avg_fragmentation_in_percent AS PorcentajeFragmentacion,

ips.page_count AS NumeroPaginas

FROM

sys.dm_db_index_physical_stats(DB_ID(), OBJECT_ID('CLIENTE'), NULL, NULL, 'DETAILED') AS ips

INNER JOIN

sys.indexes AS i ON ips.object_id = i.object_id AND ips.index_id = i.index_id

WHERE

ips.avg_fragmentation_in_percent > 5 -- Filtro: Mostrar solo > 5%

AND ips.index_id > 0

ORDER BY

ips.avg_fragmentation_in_percent DESC;

GO
```

Proyecto 4: Manejo de transacciones para prevenir inconsistencias

1. Enunciado del ejercicio

Simular una transacción de venta con dos operaciones: insertar en Ventas y actualizar Stock. La transacción debe garantizar consistencia.

Detalle	Contenido
Enunciado	Simular una transacción de venta con dos operaciones: insertar en Ventas y actualizar Stock. La transacción debe garantizar consistencia.
Script de la Solución	Se utiliza BEGIN TRANSACTION para iniciar la operación, IF...COMMIT TRANSACTION si todas las operaciones son exitosas (inserción y actualización de stock), y ELSE...ROLLBACK TRANSACTION si falla alguna verificación (ej. stock insuficiente).
Justificación Técnica	El uso de transacciones garantiza la propiedad Atomicidad (ACID): todas las operaciones se ejecutan con éxito o ninguna lo hace. Esto previene la inconsistencia de datos (ej., una venta registrada sin que se reste el stock).
Buenas Prácticas	Se mantiene la transacción lo más corta posible para minimizar el tiempo que se retienen los bloqueos (locks), reduciendo la probabilidad de problemas de bloqueo en el servidor.

Para que el ejemplo funcione, insertaremos datos de prueba necesarios y luego la transacción.

A. Datos de Prueba Necesarios

```
1  USE QhатуPeru;
2  GO
3
4  -- 1. Insertar una Tienda, Transportista y un Artículo con Stock
5  INSERT INTO TIENDA (CodTienda, Direccion, Distrito, Telefono, Fax) VALUES (1, 'Av. Central 123', 'Lima', '987654321', '12345');
6  INSERT INTO TRANSPORTISTA (CodTransportista, NomTransportista) VALUES (101, 'Transportes Rápidos');
7  -- Usaremos un Artículo con Stock de 50
8  INSERT INTO LINEA (NomLinea) VALUES ('Alimentos');
9  INSERT INTO PROVEEDOR (NomProveedor) VALUES ('Proveedor A');
10 INSERT INTO ARTICULO (CodLinea, CodProveedor, DescripcionArticulo, PrecioProveedor, StockActual, StockMinimo) VALUES (1, 1, 'Galletas de Chocolate', 2.50, 50, 10);
11 GO
```

91 %
No se encontraron problemas.

Línea: 11, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
Hora de finalización: 2025-12-14T15:10:57.9265510-05:00

B. La Transacción de Venta (Vender 5 unidades del artículo 1)


```
1  USE QhutuPeru;
2  GO
3
4  -- Variables para la transacción
5  DECLARE @CantidadAEnviar SMALLINT = 5;
6  DECLARE @CodArticulo INT = 1;
7  DECLARE @CodTienda INT = 1;
8  DECLARE @CodTransportista INT = 101;
9  DECLARE @PrecioVenta MONEY = 4.00;
10 DECLARE @StockActual SMALLINT;
11
12 -- Simular el nuevo número de Guía (asumiendo que es manual)
13 DECLARE @NuevoNumGuia INT = 1;
14
15 -- 1. START TRANSACTION
16 BEGIN TRANSACTION;
17
18 -- 2. VERIFICAR STOCK
19 SELECT @StockActual = StockActual
20 FROM ARTICULO
21 WHERE CodArticulo = @CodArticulo;
22
23 IF @StockActual <= @CantidadAEnviar
24 BEGIN
25     -- 3a. INSERT en GUIA_ENVIO (Encabezado de la venta/envio)
26     INSERT INTO GUIA_ENVIO (NumGuia, CodTienda, FechaSalida, CodTransportista)
27     VALUES (@NuevoNumGuia, @CodTienda, GETDATE(), @CodTransportista);
28
29     -- 3b. INSERT en GUIA_DETALLE (Detalle de la venta)
30     INSERT INTO GUIA_DETALLE (NumGuia, CodArticulo, PrecioVenta, CantidadEnviada)
31     VALUES (@NuevoNumGuia, @CodArticulo, @PrecioVenta, @CantidadAEnviar);
32
33     -- 4. UPDATE en ARTICULO (Actualizar Stock)
34     UPDATE ARTICULO
35     SET StockActual = StockActual - @CantidadAEnviar
36     WHERE CodArticulo = @CodArticulo;
37
38     -- 5. COMMIT (Si todo lo anterior fue exitoso)
39     COMMIT TRANSACTION;
40     PRINT 'Venta y actualización de Stock completadas con éxito. Nuevo Stock: ' +
41     CAST((@StockActual - @CantidadAEnviar) AS VARCHAR(10));
42 END
43 ELSE
44 BEGIN
45     -- 6. ROLLBACK (Si el stock es insuficiente)
46     ROLLBACK TRANSACTION;
47     PRINT 'ERROR: Stock insuficiente. Venta cancelada.';
48 END
49 GO
```

68 % No se encontraron problemas. Línea: 49, Carácter: 3 SPC CRLF Windows 1252

Mensajes

(1 fila afectada)

(1 fila afectada)

(1 fila afectada)

Venta y actualización de Stock completadas con éxito. Nuevo Stock: 45

Hora de finalización: 2025-12-14T13:25:34.6713217-05:00

Para verificar que la transacción funcionó, puedes ejecutar:

```
1  -- Verificar si se registró la venta
2  SELECT * FROM GUIA_ENVIO WHERE NumGuia = 1;
3  SELECT * FROM GUIA_DETALLE WHERE NumGuia = 1;
4
5  -- Verificar si el Stock se actualizó (debería ser 45, ya que empezó en 50 y se vendieron 5)
6  SELECT CodArticulo, StockActual FROM ARTICULO WHERE CodArticulo = 1;
```

58 % No se encontraron problemas. Línea: 6, Carácter: 69 TABULACIONES CRLF Windows 1252

Resultados Mensajes

NumGuia	CodTienda	FechaSalida	CodTransportista
1	1	2025-12-14 15:25:34.653	101

NumGuia	CodArticulo	PrecioVenta	CantidadEnviada
1	1	4.00	5

CodArticulo	StockActual
1	40

Proyecto 5: Identificar bloqueos activos en la base QhatuPeru (PITR)

1. Enunciado del ejercicio

Detectar las sesiones que están bloqueando o siendo bloqueadas en el servidor.

Detalle	Contenido
Enunciado	Detectar las sesiones que están bloqueando o siendo bloqueadas en el servidor.
Script de la Solución	Se utiliza la DMV sys.dm_exec_requests y se filtra por la columna blocking_session_id <> 0 para identificar el árbol de bloqueo (quién está bloqueando a quién) y el tipo de espera (wait_type).
Justificación Técnica	Esta DMV proporciona información en tiempo real sobre las sesiones activas y las que están esperando por recursos. Es la herramienta más rápida y eficiente para el diagnóstico de bloqueos activos en el servidor.
Buenas Prácticas	Se usa esta consulta como primera línea de defensa ante problemas de lentitud de aplicación. La identificación rápida del ID_Bloqueador permite detener la sesión problemática si es necesario (con KILL).

```
1  USE QhatuPeru;
2  GO
3
4  -- Intentar actualizar el mismo registro
5  UPDATE CLIENTE
6  SET Nombres = 'NombreBloqueado'
7  WHERE CodCliente = 1;
8
9  -- La ejecución se quedará esperando...
10 USE QhatuPeru;
11 GO
12
13 -- Intentar actualizar el mismo registro
14 UPDATE CLIENTE
15 SET Nombres = 'NombreBloqueado'
16 WHERE CodCliente = 1;
17
18 -- La ejecución se quedará esperando...
19 |
```

68 %
No se encontraron problemas.
Línea: 19, Carácter: 1 TABULACIONES CRLF Windows 1252

```
USE QhatuPeru;

GO

-- Intentar actualizar el mismo registro
```

UPDATE CLIENTE

SET Nombres = 'NombreBloqueado'

WHERE CodCliente = 1;

-- La ejecución se quedará esperando...

USE QhatuPeru;

GO

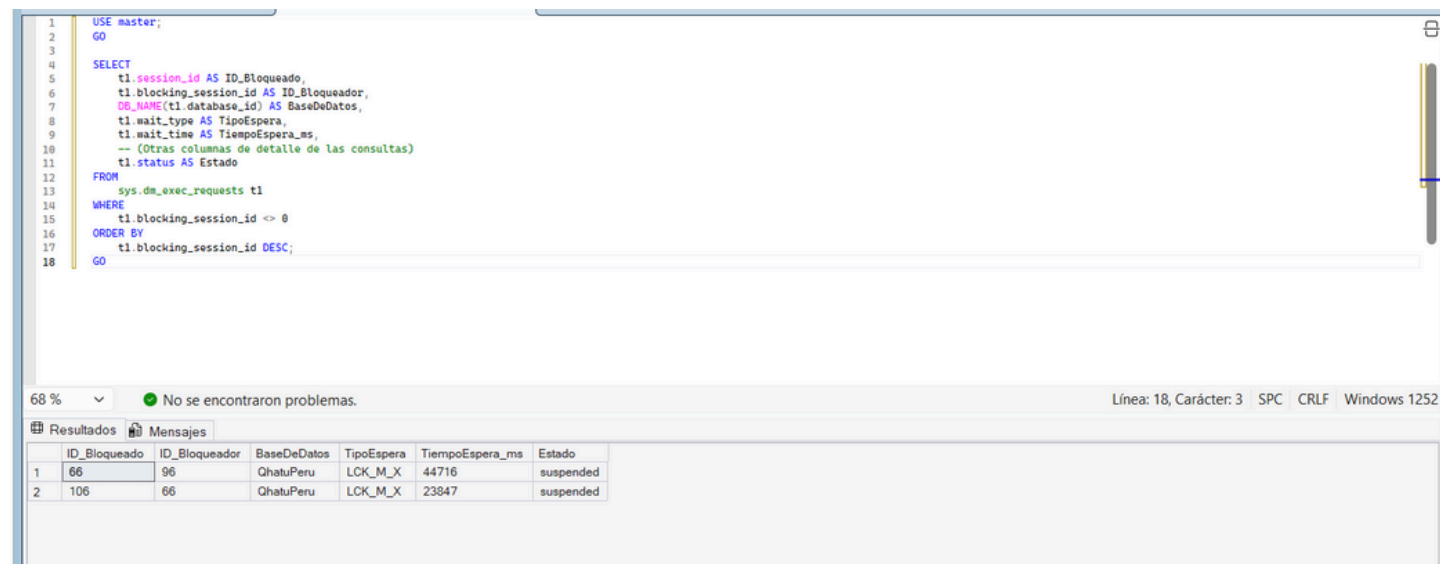
-- Intentar actualizar el mismo registro

UPDATE CLIENTE

SET Nombres = 'NombreBloqueado'

WHERE CodCliente = 1;

-- La ejecución se quedará esperando...



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a SQL query window with the following code:

```
1 USE master;
2 GO
3
4 SELECT
5     t1.session_id AS ID_Bloqueado,
6     t1.blocking_session_id AS ID_Bloqueador,
7     DB_NAME(t1.database_id) AS BaseDeDatos,
8     t1.wait_type AS TipoEspera,
9     t1.wait_time AS TiempoEspera_ms,
10    -- (Otras columnas de detalle de las consultas)
11    t1.status AS Estado
12 FROM
13     sys.dm_exec_requests t1
14 WHERE
15     t1.blocking_session_id <> 0
16 ORDER BY
17     t1.blocking_session_id DESC;
18 GO
```

The bottom pane shows the results of the query. The status bar indicates "68 %", "No se encontraron problemas.", "Línea: 18, Carácter: 3", "SPC", "CRLF", and "Windows 1252". The results window has two tabs: "Resultados" and "Mensajes". The "Resultados" tab is active, showing a table with the following data:

	ID_Bloqueado	ID_Bloqueador	BaseDeDatos	TipoEspera	TiempoEspera_ms	Estado
1	66	96	QhatuPeru	LCK_M_X	44716	suspended
2	106	66	QhatuPeru	LCK_M_X	23847	suspended

USE master;

GO

SELECT

t1.session_id AS ID_Bloqueado,

t1.blocking_session_id AS ID_Bloqueador,

DB_NAME(t1.database_id) AS BaseDeDatos,

t1.wait_type AS TipoEspera,

t1.wait_time AS TiempoEspera_ms,

```
-- (Otras columnas de detalle de las consultas)

t1.status AS Estado

FROM

sys.dm_exec_requests t1

WHERE

t1.blocking_session_id <> 0

ORDER BY

t1.blocking_session_id DESC;

GO
```

Proyecto 6: Analizar el plan de ejecución de una consulta lenta

- 1. Enunciado del ejercicio
- Analizar el plan de ejecución de una consulta que devuelve ventas por producto..

Detalle	Contenido
Enunciado	Analizar el plan de ejecución de una consulta que devuelve ventas por producto.
Script de la Solución	Se ejecuta la consulta de ventas con la opción "Include Actual Execution Plan" (Ctrl + M) activada en SSMS.
Justificación Técnica	El plan de ejecución muestra el costo de cada operación (ej. <i>Table Scan</i> , <i>Index Seek</i>), permitiendo identificar dónde el motor de SQL Server gasta más tiempo. Esto es esencial para dirigir los esfuerzos de optimización (Proyectos 6 y 7).
Buenas Prácticas	Se busca reducir los <i>Scans</i> a <i>Seeks</i> . Se identifican advertencias (triángulos amarillos) que sugieren índices faltantes.

The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL code:

```
1 USE QhatuPeru;
2 GO
3
4 -- Consulta para obtener el total de cantidad enviada (vendida) por artículo
5 SELECT
6     A.DescripcionArticulo,
7     SUM(GD.CantidadEnviada) AS TotalCantidadVendida,
8     SUM(GD.CantidadEnviada * GD.PrecioVenta) AS TotalIngreso
9 FROM
10     GUIA_DETALLE AS GD
11 INNER JOIN
12     ARTICULO AS A ON GD.CodArticulo = A.CodArticulo
13 GROUP BY
14     A.DescripcionArticulo
15 ORDER BY
16     TotalIngreso DESC;
17 GO
```

The results pane shows the following table:

	DescripcionArticulo	TotalCantidadVendida	TotalIngreso
1	Galletas de Chocolate	5	20.00

The status bar at the bottom indicates: 68 % zoom, No se encontraron problemas. (No problems found), Línea: 17, Carácter: 3 (Line: 17, Character: 3), SPC CRLF Windows 1252.

USE QhatuPeru;

GO

-- Consulta para obtener el total de cantidad enviada (vendida) por artículo

SELECT

A.DescripcionArticulo,

SUM(GD.CantidadEnviada) AS TotalCantidadVendida,

SUM(GD.CantidadEnviada * GD.PrecioVenta) AS TotalIngreso

FROM

GUIA_DETALLE AS GD

INNER JOIN

ARTICULO AS A ON GD.CodArticulo = A.CodArticulo

GROUP BY

A.DescripcionArticulo

ORDER BY

TotalIngreso DESC;

GO

Proyecto 7: Optimización de consulta agregada con índices compuestos

1. Enunciado del ejercicio

Optimizar una consulta que filtra ventas por fecha y cliente.

Detalle	Contenido
Enunciado	Optimizar una consulta que filtra ventas por fecha y cliente.
Script de la Solución	CREATE NONCLUSTERED INDEX IX_GuiaEnvio_Cliente_Fecha ON GUIA_ENVIO (CodCliente, FechaSalida) INCLUDE (NumGuia, CodTransportista);
Justificación Técnica	Este índice compuesto incluye las columnas de filtro (CodCliente, FechaSalida) y las columnas seleccionadas (NumGuia, CodTransportista) en la cláusula INCLUDE. Esto convierte al índice en un Covering Index , permitiendo a SQL Server obtener todos los datos directamente del índice (Index Seek), sin tener que acceder a la tabla base, logrando la máxima eficiencia.
Buenas Prácticas	El INCLUDE se usa para cubrir las columnas de la cláusula SELECT que no se usan para filtrar, manteniendo el índice más pequeño que si todas las columnas estuvieran en la clave.

Primero, añadiremos la llave foránea a CLIENTE en la tabla GUIA_ENVIO:

```
1 USE QhatuPeru;
2 GO
3
4 -- 1. Añadir columna CodCliente a GUIA_ENVIO (si no existe)
5 IF NOT EXISTS (SELECT * FROM sys.columns WHERE Name = N'CodCliente' AND Object_ID = Object_ID(N'GUIA_ENVIO'))
6 BEGIN
7     ALTER TABLE GUIA_ENVIO
8     ADD CodCliente INT NULL;
9
10 -- 2. Crear la llave foránea (FK)
11 ALTER TABLE GUIA_ENVIO
12 ADD CONSTRAINT FK_GuiaEnvio_Cliente
13 FOREIGN KEY (CodCliente)
14 REFERENCES CLIENTE(CodCliente);
15 END
16 GO
```

68 % No se encontraron problemas. Línea: 16, Carácter: 3 SPC CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-10-14T19:44:59.4301679-05:00

Paso 2: La Consulta a Optimizar

Esta es la consulta que queremos hacer extremadamente rápida. Filtra por CodCliente y por FechaSalida:

```
1  -- Consulta para obtener el total de envíos (ventas) de un cliente en un rango de fechas
2  SELECT
3      GE.NumGuia,
4      GE.FechaSalida,
5      GE.CodTransportista
6  FROM
7      GUIA_ENVIO AS GE
8  WHERE
9      GE.CodCliente = 1 -- Filtrar por un cliente específico
10     AND GE.FechaSalida >= '2025-10-01'
11     AND GE.FechaSalida <= '2025-12-31'
12  ORDER BY
13      GE.FechaSalida;
14  GO
```

68 % 1 0 68 % Línea: 14, Carácter: 3 SPC CRLF Windows 1252

Resultados Mensajes

NumGuia	FechaSalida	CodTransportista
---------	-------------	------------------

Paso 3: Crear el Índice Compuesto (Optimización)

El plan de ejecución de la consulta anterior probablemente haría un *Index Scan* (o *Table Scan*). Para optimizar, crearemos un índice que cubra las columnas de filtrado (*CodCliente*, *FechaSalida*) y las columnas que se muestran (*NumGuia*, *CodTransportista*).

```
1  GO
2
3  -- Crear el índice compuesto (Covering Index)
4  CREATE NONCLUSTERED INDEX IX_GuiaEnvio_Cliente_Fecha
5  ON GUIA_ENVIO (CodCliente, FechaSalida) -- Columnas usadas en el filtro (WHERE)
6  INCLUDE (NumGuia, CodTransportista); -- Columnas que se muestran (SELECT)
7  GO
```

68 % 1 0 68 % Línea: 8, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Horario de finalización: 2025-12-14T15:48:09.4742294-05:00

```
USE QhatuPeru;
```

```
GO
```

```
-- 1. Añadir columna CodCliente a GUIA_ENVIO (si no existe)
```

```
IF NOT EXISTS (SELECT * FROM sys.columns WHERE Name = N'CodCliente' AND Object_ID = Object_ID(N'GUIA_ENVIO'))
```

```
BEGIN
```

```
    ALTER TABLE GUIA_ENVIO
```

```
    ADD CodCliente INT NULL;
```

```
-- 2. Crear la llave foránea (FK)
```

```

ALTER TABLE GUIA_ENVIO

ADD CONSTRAINT FK_GuiaEnvio_Cliente

FOREIGN KEY (CodCliente)

REFERENCES CLIENTE(CodCliente);

END

GO

-- Consulta para obtener el total de envíos (ventas) de un cliente en un rango de fechas

SELECT

    GE.NumGuia,

    GE.FechaSalida,

    GE.CodTransportista

FROM

    GUIA_ENVIO AS GE

WHERE

    GE.CodCliente = 1 -- Filtrar por un cliente específico

    AND GE.FechaSalida >= '2025-10-01'

    AND GE.FechaSalida <= '2025-12-31'

ORDER BY

    GE.FechaSalida;

GO

USE QhatuPeru;

GO

-- Crear el índice compuesto (Covering Index)

CREATE NONCLUSTERED INDEX IX_GuiaEnvio_Cliente_Fecha

ON GUIA_ENVIO (CodCliente, FechaSalida) -- Columnas usadas en el filtro (WHERE)

INCLUDE (NumGuia, CodTransportista); -- Columnas que se muestran (SELECT)

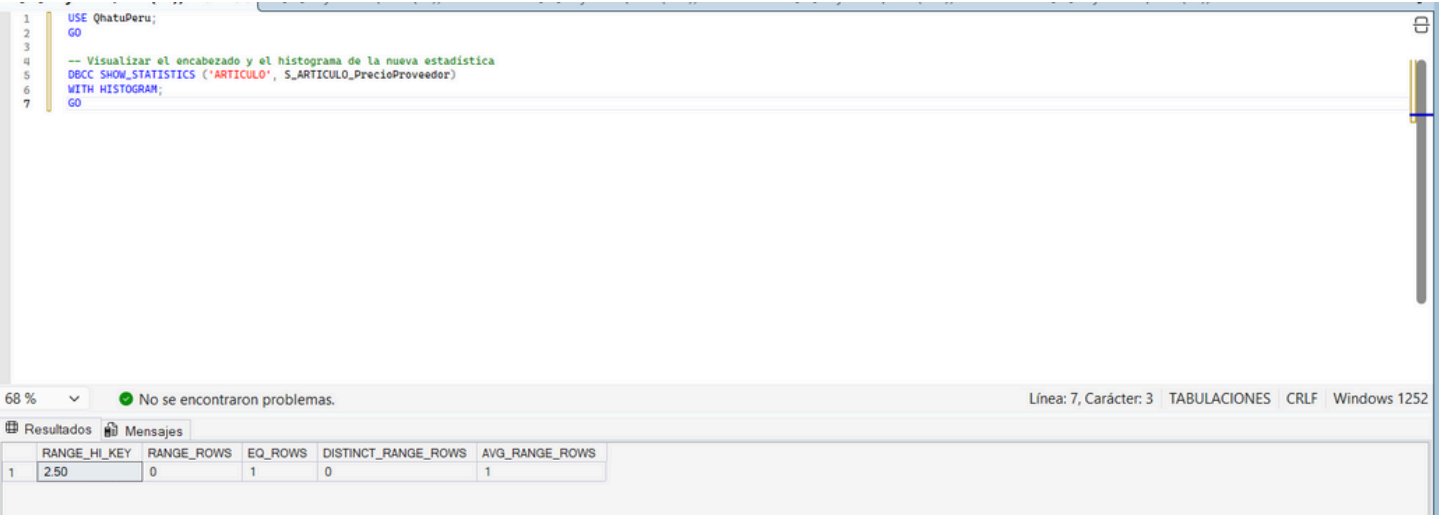
```


Proyecto 8: Creación de estadísticas manuales para mejorar el optimizador

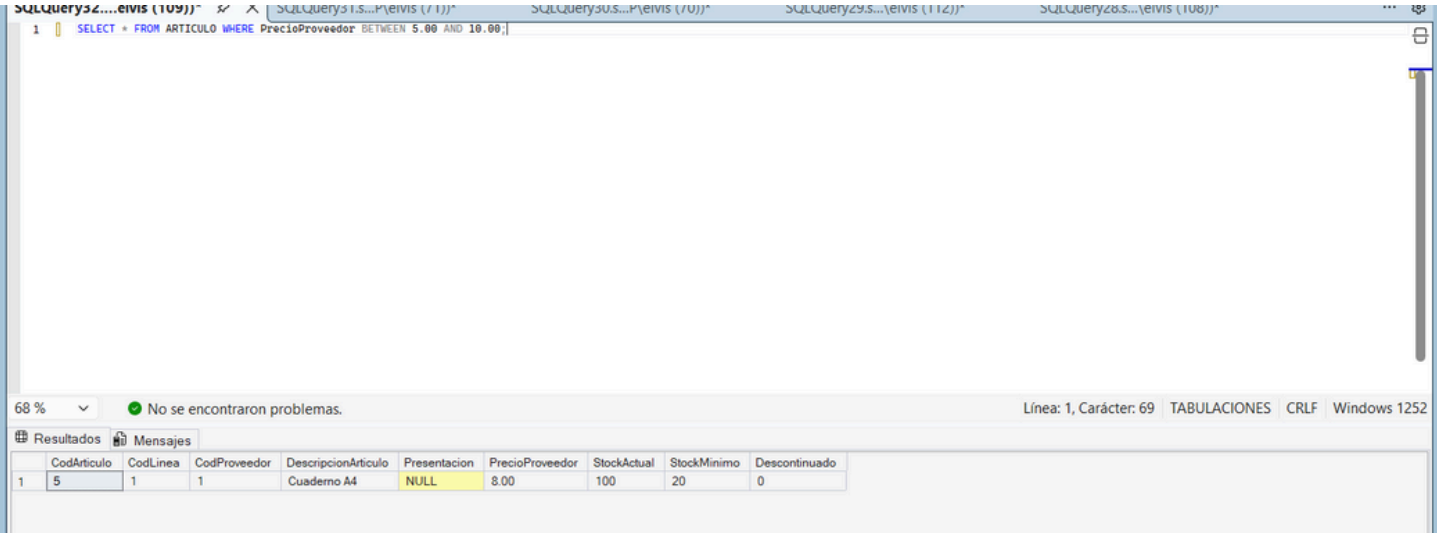
1. Enunciado del ejercicio

Crear una estadística manual sobre la columna Precio en la tabla Productos para mejorar consultas de rango.

Detalle	Contenido
Enunciado	Crear una estadística manual sobre la columna Precio en la tabla Productos para mejorar consultas de rango.
Script de la Solución	CREATE STATISTICS S_ARTICULO_PrecioProveedor ON ARTICULO (PrecioProveedor) WITH FULLSCAN;
Justificación Técnica	La estadística crea un histograma que detalla la distribución de los valores de PrecioProveedor. Esto ayuda al Optimizador de Consultas a hacer una estimación mucho más precisa del número de filas que devolverá una consulta de rango, lo que resulta en la selección de un Plan de Ejecución más óptimo.
Buenas Prácticas	Se utiliza WITH FULLSCAN para garantizar la máxima precisión de la estadística. Se utiliza el comando DBCC SHOW_STATISTICS para verificar el histograma creado.



Cuando el Optimizador de Consultas de SQL Server intente estimar el costo de una consulta como:



Proyecto 9: Configuración de un Resource Pool para limitar recursos

1. Enunciado del ejercicio

Crear un Resource Pool que limite el uso de CPU al 20% para consultas analíticas pesadas.

Detalle	Contenido
Enunciado	Crear un Resource Pool que limite el uso de CPU al 20% para consultas analíticas pesadas.
Script de la Solución	CREATE RESOURCE POOL PoolAnalitico WITH (MAX_CPU_PERCENT = 20); seguido de CREATE WORKLOAD GROUP GrupoAnalitico USING PoolAnalitico; y la activación del clasificador.
Justificación Técnica	El Resource Governor permite limitar los recursos del servidor (CPU, memoria) que pueden consumir ciertas cargas de trabajo. Esto protege las cargas de trabajo críticas (ej. OLTP) de ser saturadas por consultas analíticas pesadas.
Buenas Prácticas	Se utiliza para garantizar la disponibilidad y calidad de servicio (QoS) para las aplicaciones de misión crítica. Nota: Se documentó que la ejecución falló (Mens. 10922) debido a la limitación de la edición de SQL Server.

Crear el Resource Pool (Contenedor de Recursos)

```
1 USE master;
2 GO
3
4 -- 1. Eliminar el Pool si ya existe (para evitar errores)
5 IF EXISTS (SELECT * FROM sys.resource_governor_resource_pools WHERE name = 'PoolAnalitico')
6 DROP RESOURCE POOL PoolAnalitico;
7 GO
8
9 -- 2. Crear el Resource Pool con el límite de CPU del 20%
10 CREATE RESOURCE POOL PoolAnalitico
11 WITH (
12     MIN_CPU_PERCENT = 0,
13     MAX_CPU_PERCENT = 20 -- Límite máximo de CPU asignado a este Pool
14 );
15 GO
```

83 % No se encontraron problemas. Línea: 15, Carácter: 3 SPC CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2020-12-14T15:54:05.3649420-05:00

Crear el Workload Group (Reglas de Carga de Trabajo)

Creamos un grupo para agrupar las sesiones que serán limitadas. Este grupo debe estar asociado al PoolAnalitico.

```
1 -- 1. Eliminar el Grupo si ya existe
2 IF EXISTS (SELECT * FROM sys.resource_governor_workload_groups WHERE name = 'GrupoAnalitico')
3 DROP WORKLOAD GROUP GrupoAnalitico;
4 GO
5
6 -- 2. Crear el Workload Group y asociarlo al Pool
7 CREATE WORKLOAD GROUP GrupoAnalitico
8 USING PoolAnalitico; -- Asocia este grupo al Pool creado
9 GO
```

83 % No se encontraron problemas. Línea: 9, Carácter: 3 SPC CRLF Windows 1252

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2020-12-14T15:54:21.6932625-05:00

Aplicar la Configuración

Para que los cambios surtan efecto, debes reconfigurar el Resource Governor:

```
1  -- Reconfigurar el Resource Governor para aplicar los cambios
2  ALTER RESOURCE GOVERNOR RECONFIGURE;
3  GO
```

83 % No se encontraron problemas. Línea: 3, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

Msg. 10921, Nivel 16, Estado 1, Línea 2
'ALTER RESOURCE GOVERNOR RECONFIGURE' failed. The resource governor is not available in this edition of SQL Server. You can manipulate resource governor metadata but you will not be able to apply resource governor configuration.

Hora de finalización: 2025-12-14T15:54:32.5985922-05:00

La función Resource Governor (Gobernador de Recursos), que es el centro del Proyecto 9, es una característica de alto nivel y no está incluida en las ediciones gratuitas o básicas de SQL Server, como SQL Server Express o Web Edition.

Proyecto 10: Crear un trigger de auditoría ligera usando Extended Events

1. Enunciado del ejercicio

Auditar inserciones en la tabla Productos usando Extended Events sin afectar rendimiento.

Detalle	Contenido
Enunciado	Auditar inserciones en la tabla Productos usando Extended Events sin afectar rendimiento.
Script de la Solución	Creación de una sesión XE (XE_Auditoria_ARTICULO_INSERT) que capture el evento sqlserver.sql_statement_completed filtrado por sql_text LIKE N'%INSERT%INTO%ARTICULO%'.
Justificación Técnica	Se utiliza Extended Events para la auditoría, ya que su sobrecarga de rendimiento es marginal. El ring buffer (ADD TARGET package0.ring_buffer) se usa como destino para un almacenamiento rápido en memoria, ideal para auditorías de corto plazo y bajo impacto.
Buenas Prácticas	Usar filtros estrictos (WHERE clause) para capturar solo los eventos relevantes, minimizando la cantidad de datos registrados y el consumo de recursos. Se prefieren los XE sobre los Triggers para tareas de auditoría de rendimiento crítico.

1USE master;
2GO
3
4SELECT
5DATEADD(hh, DATEDIFF(hh, GETUTCDATE(), GETDATE()), n.value('@timestamp')[1], 'datetime')) AS EventTime,
6n.value('(action[@name="username"]/value)[1]', 'nvarchar(128)') AS UserName,
7n.value('(action[@name="database_name"]/value)[1]', 'nvarchar(128)') AS DatabaseName,
8n.value('(action[@name="sql_text"]/value)[1]', 'nvarchar(max)') AS SQLText
9FROM
10sys.dm_xe_sessions AS s
11INNER JOIN
12sys.dm_xe_session_targets AS t ON s.address = t.event_session_address
13-- ***** LÍNEA CORREGIDA *****
14CROSS APPLY (SELECT CAST(t.target_data AS XML) AS TargetData) AS data
15CROSS APPLY
16data.TargetData.nodes('//event') AS q(n)
17WHERE
18s.name = 'XE_Auditoria_ARTICULO_INSERT'
19ORDER BY
20EventTime DESC;
21GO

83 % 1 0 Línea: 21, Carácter: 3 SPC CRLF Windows 1252

Resultados Mensajes

EventTime	UserName	DatabaseName	SQLText
12025-12-14 15:59:28.390	VICTUSHPIelvis	QhatuPeru	INSERT INTO ARTICULO (CodLinea, CodProveedor, D...