GUÍA RÁPIDA SQL SERVER / SOLUCIÓN DE PROBLEMAS / INTERMEDIO

Encontrar bloqueos en SQL Server

Identifica sesiones bloqueadas, bloqueadores, esperas de lock y el bloqueador principal para diagnosticar bloqueos en SQL Server sin terminar la sesión equivocada.



Actualizado 4 oct 2026 Versión en línea →

PROBLEMA

Los usuarios informan que SQL Server está lento o «colgado», pero el uso de CPU no es necesariamente alto. Puede que una o varias sesiones estén esperando locks que mantiene otra sesión.

El objetivo es responder:

- ¿Qué sesiones están bloqueadas?
- ¿Qué sesión las está bloqueando?
- ¿Qué SQL está ejecutando cada lado?
- ¿Qué recurso se está esperando?
- ¿Hay una cadena de bloqueos?
- ¿Qué sesión es el bloqueador principal?

La consulta bloqueada suele ser la **víctima**, no la causa raíz.

RESPUESTA RÁPIDA

Empieza con `sys.dm_exec_requests` y busca solicitudes activas con `blocking_session_id > 0`.

SQL

```sql
SELECT
    r.session_id AS blocked_session_id,
    r.blocking_session_id,
    DB_NAME(r.database_id) AS database_name,
    r.status,
    r.command,
    r.wait_type,
    r.wait_time AS wait_ms,
    r.wait_resource,
    CAST(r.total_elapsed_time / 1000.0 AS decimal(18,1)) AS elapsed_seconds,
    s.login_name,
    s.host_name,
    s.program_name,
    txt.text AS blocked_batch_text
FROM sys.dm_exec_requests AS r
INNER JOIN sys.dm_exec_sessions AS s
    ON s.session_id = r.session_id
OUTER APPLY sys.dm_exec_sql_text(r.sql_handle) AS txt
WHERE
    r.blocking_session_id > 0
ORDER BY
    r.wait_time DESC;
```

> Si esta consulta devuelve filas, hay bloqueos **en este momento**.
>
> El siguiente paso no es terminar la sesión bloqueada. Encuentra el bloqueador y determina si está al comienzo de la cadena.

Código / comandos

1. Mostrar las solicitudes bloqueadas y sus bloqueadores

Usa esta consulta como primera comprobación de bloqueos.

```sql
SELECT
    r.session_id AS blocked_session_id,
    r.blocking_session_id,
    DB_NAME(r.database_id) AS database_name,
    r.status,
    r.command,
    r.wait_type,
    r.wait_time AS wait_ms,
    r.wait_resource,
    CAST(r.total_elapsed_time / 1000.0 AS decimal(18,1)) AS elapsed_seconds,
    s.login_name,
    s.host_name,
    s.program_name,
    txt.text AS blocked_batch_text
FROM sys.dm_exec_requests AS r
INNER JOIN sys.dm_exec_sessions AS s
    ON s.session_id = r.session_id
OUTER APPLY sys.dm_exec_sql_text(r.sql_handle) AS txt
WHERE
    r.blocking_session_id > 0
ORDER BY
    r.wait_time DESC;
```

Un `blocking_session_id` positivo significa que otra sesión está identificada como el bloqueador.

No uses `blocking_session_id <> 0` como regla general para el análisis habitual de bloqueos. SQL Server puede devolver valores negativos especiales en algunos casos internos o distribuidos.

2. Mostrar en paralelo el SQL del bloqueador y el de la sesión bloqueada

Suele ser la consulta más útil en producción, porque da contexto de ambos lados.

```sql
SELECT
    blocked.session_id AS blocked_session_id,
    blocked.blocking_session_id,
    DB_NAME(blocked.database_id) AS database_name,
    blocked.status AS blocked_status,
    blocked.wait_type,
    blocked.wait_time AS blocked_wait_ms,
    blocked.wait_resource,
    blocked_sql.text AS blocked_batch_text,

    blocker.status AS blocker_status,
    blocker.command AS blocker_command,
    blocker.cpu_time AS blocker_cpu_ms,
    blocker.logical_reads AS blocker_logical_reads,
    blocker.start_time AS blocker_start_time,
```

```
    blocker_sql.text AS blocker_batch_text,

    blocker_session.login_name AS blocker_login_name,
    blocker_session.host_name AS blocker_host_name,
    blocker_session.program_name AS blocker_program_name,
    blocker_session.open_transaction_count AS blocker_open_transactions
FROM sys.dm_exec_requests AS blocked
LEFT JOIN sys.dm_exec_requests AS blocker
    ON blocker.session_id = blocked.blocking_session_id
LEFT JOIN sys.dm_exec_sessions AS blocker_session
    ON blocker_session.session_id = blocked.blocking_session_id
OUTER APPLY sys.dm_exec_sql_text(blocked.sql_handle) AS blocked_sql
OUTER APPLY sys.dm_exec_sql_text(blocker.sql_handle) AS blocker_sql
WHERE
    blocked.blocking_session_id > 0
ORDER BY
    blocked.wait_time DESC;
```

Advertencia importante: la sesión bloqueadora puede estar **inactiva** (sleeping) y, por lo tanto, no tener ninguna fila en `sys.dm_exec_requests`.

Esto ocurre a menudo cuando una aplicación inicia una transacción, modifica datos y luego deja la transacción abierta sin una solicitud activa.

En ese caso, revisa `sys.dm_exec_sessions` y las DMV de transacciones en lugar de suponer que el bloqueador desapareció.

---

3. Encontrar los probables bloqueadores principales

En la práctica, un bloqueador principal es una sesión que bloquea a otras pero que no está bloqueada en ese momento por otro ID de sesión positivo.

SQL

```
WITH blocked AS
(
    SELECT
        r.session_id,
        r.blocking_session_id,
        r.wait_type,
        r.wait_time
    FROM sys.dm_exec_requests AS r
    WHERE r.blocking_session_id > 0
),
blockers AS
(
    SELECT DISTINCT
        blocking_session_id AS session_id
    FROM blocked
)
SELECT
    b.session_id AS head_blocker_session_id,
    s.status AS session_status,
    s.login_name,
    s.host_name,
    s.program_name,
    s.open_transaction_count,
    r.status AS request_status,
    r.command,
    r.wait_type,
    r.blocking_session_id,
```

```
    txt.text AS batch_text
FROM blockers AS b
INNER JOIN sys.dm_exec_sessions AS s
    ON s.session_id = b.session_id
LEFT JOIN sys.dm_exec_requests AS r
    ON r.session_id = b.session_id
OUTER APPLY sys.dm_exec_sql_text(r.sql_handle) AS txt
WHERE
    NOT EXISTS
    (
        SELECT 1
        FROM blocked AS x
        WHERE
            x.session_id = b.session_id
            AND x.blocking_session_id > 0
    )
ORDER BY
    b.session_id;
```

Úsala para acotar la investigación a la sesión que está al comienzo de una cadena de bloqueos.

Es una consulta práctica de triaje, no un grafo de locks completo.

4. Mostrar las tareas en espera de las sesiones bloqueadas

Usa `sys.dm_os_waiting_tasks` cuando necesites más detalle sobre la espera actual y el recurso.

```sql
SELECT
    wt.session_id,
    wt.blocking_session_id,
    wt.wait_type,
    wt.wait_duration_ms,
    wt.resource_description
FROM sys.dm_os_waiting_tasks AS wt
WHERE
    wt.blocking_session_id IS NOT NULL
ORDER BY
    wt.wait_duration_ms DESC;
```

Esto puede revelar esperas a nivel de tarea, más detalladas que la instantánea a nivel de solicitud.

Una consulta paralela puede tener varias tareas esperando recursos distintos, así que no supongas que una fila siempre representa toda la solicitud.

5. Correlacionar las esperas bloqueadas con la información de locks

Usa esta consulta solo cuando necesites detalle a nivel de lock.

```sql
SELECT
    wt.session_id AS blocked_session_id,
    wt.blocking_session_id,
    wt.wait_type,
    wt.wait_duration_ms,
    wt.resource_description,

    l.resource_type,
    l.resource_database_id,
    DB_NAME(l.resource_database_id) AS database_name,
```

```
    l.resource_associated_entity_id,
    l.request_mode,
    l.request_status,
    l.request_session_id
FROM sys.dm_os_waiting_tasks AS wt
INNER JOIN sys.dm_tran_locks AS l
    ON l.lock_owner_address = wt.resource_address
WHERE
    wt.blocking_session_id IS NOT NULL
ORDER BY
    wt.wait_duration_ms DESC;
```

Úsala cuando necesites entender el recurso realmente bloqueado o el modo de lock implicado.

`sys.dm_tran_locks` puede devolver una gran cantidad de datos en una instancia con mucha actividad, así que evita consultarla repetidamente sin motivo.

6. Comprobar si un bloqueador tiene una transacción abierta

Una sesión inactiva con una transacción abierta es una causa habitual de bloqueos persistentes.

SQL

```
DECLARE @session_id int = 57;

SELECT
    s.session_id,
    s.status,
    s.login_name,
    s.host_name,
    s.program_name,
    s.open_transaction_count,
    at.transaction_begin_time,
    at.transaction_type,
    at.transaction_state
FROM sys.dm_exec_sessions AS s
LEFT JOIN sys.dm_tran_session_transactions AS st
    ON st.session_id = s.session_id
LEFT JOIN sys.dm_tran_active_transactions AS at
    ON at.transaction_id = st.transaction_id
WHERE
    s.session_id = @session_id;
```

Si el bloqueador tiene una transacción abierta desde hace tiempo y ninguna solicitud activa, investiga el ciclo de vida de las transacciones de la aplicación antes de actuar.

7. Revisar el último comando enviado por un bloqueador inactivo

Cuando un bloqueador no tiene una solicitud activa, `sys.dm_exec_requests` no puede darte el texto SQL actual.

Para una investigación puntual, `DBCC INPUTBUFFER` puede mostrar la última instrucción enviada por la sesión.

SQL

```
DBCC INPUTBUFFER (57);
```

Usa el ID de sesión real del bloqueador.

Trata el resultado como contexto, no como prueba de que la instrucción mostrada sea exactamente la que adquirió cada uno de los locks que siguen retenidos.

Qué significa el resultado

Campo	Por qué importa
`blocked_session_id`	Sesión que está esperando por culpa de otra sesión.
`blocking_session_id`	Sesión que SQL Server identifica como el bloqueador de la solicitud actual.
`wait_type`	Espera actual. Las esperas de lock suelen empezar por `LCK_M_`.
`wait_time` / `wait_duration_ms`	Cuánto tiempo lleva esperando la solicitud o tarea actual.
`wait_resource` / `resource_description`	Recurso implicado en la espera. Útil para un análisis más profundo de locks.
`open_transaction_count`	Ayuda a identificar sesiones que mantienen transacciones abiertas.
`program_name`	Aplicación cliente que informa la conexión. Útil para encontrar el servicio o job responsable.
`host_name`	Nombre del equipo cliente que informa la conexión.
`request_mode`	Modo de lock solicitado o concedido, como los modos compartido o exclusivo.
`request_status`	Estado de la solicitud de lock, por ejemplo `GRANTED`, `WAIT` o `CONVERT`.

Qué revisar

Muchas sesiones bloqueadas por una sola sesión

Es una cadena de bloqueos clásica.

Prioriza el bloqueador común en lugar de examinar cada sesión bloqueada por separado.

Tipos de espera `LCK_M_*`

Indican esperas de lock.

Algunos ejemplos son las esperas de locks compartidos, exclusivos, de actualización, de esquema y de intención.

El tipo exacto de espera de lock ayuda a caracterizar la contención, pero la causa raíz sigue requiriendo contexto de la transacción y de la carga de trabajo.

Bloqueador inactivo con transacciones abiertas

Este caso es especialmente importante.

Una sesión puede estar inactiva desde el punto de vista de la aplicación y, aun así, mantener locks porque su transacción no se ha confirmado ni revertido.

Investiga:

- la antigüedad de la transacción
- el manejo de transacciones de la aplicación
- el comportamiento del pool de conexiones
- las rutas de error que pueden omitir el commit o el rollback

Esperas largas con SQL bloqueador corto

Una instrucción sencilla puede mantener locks durante mucho tiempo si se ejecuta dentro de una transacción grande o de larga duración.

No juzgues el impacto de un bloqueador solo por el texto de la instrucción visible.

Bloqueos repetidos desde la misma aplicación

Revisa:

- `login_name`
- `host_name`
- `program_name`

Los bloqueos recurrentes de una misma aplicación o job suelen requerir una corrección en la aplicación, la consulta o la transacción, en lugar de intervenciones manuales repetidas.

> **NOTAS DE PRODUCCIÓN**
>
> #### Los bloqueos son normales; el problema son los bloqueos prolongados
>
> Los locks son una parte esencial del control de concurrencia de SQL Server.
>
> Una sesión bloqueada no indica automáticamente un defecto. Céntrate en los bloqueos prolongados, generalizados, inesperados o que afectan a los usuarios.
>
> #### Los datos de bloqueos son transitorios
>
> `sys.dm_exec_requests` y `sys.dm_os_waiting_tasks` muestran la actividad actual.
>
> Si el bloqueo desaparece antes de que lo captures, la evidencia desaparece con él.
>
> Para incidentes intermitentes, captura instantáneas de las DMV a lo largo del tiempo o usa Extended Events o una herramienta de monitoreo.
>
> #### Los permisos varían según la versión de SQL Server
>
> En SQL Server 2019 y versiones anteriores, ver todas las sesiones y solicitudes suele requerir `VIEW SERVER STATE`.
>
> En SQL Server 2022 y versiones posteriores, el permiso equivalente a nivel de servidor es `VIEW SERVER PERFORMANCE STATE`.
>
> Azure SQL Database tiene reglas de ámbito y permisos diferentes.
>
> #### Evita recorridos amplios de `sys.dm_tran_locks` salvo que sea necesario
>
> Los diagnósticos a nivel de lock pueden devolver un conjunto de resultados grande en un servidor con mucha actividad.
>
> Empieza por las solicitudes y las tareas en espera, y pasa a `sys.dm_tran_locks` cuando el detalle a nivel de recurso sea realmente útil.
>
> #### No termines sesiones a la ligera
>
> Terminar un bloqueador puede:
>
> - revertir (rollback) una transacción grande
> - generar bloqueos adicionales durante el rollback
> - provocar errores en la aplicación
> - interrumpir procesos de negocio
> - ocultar la verdadera causa raíz
>
> Si terminar la sesión es necesario desde el punto de vista operativo, entiende primero la transacción y el impacto del rollback.

ERRORES COMUNES

1. Terminar la sesión bloqueada

La sesión bloqueada normalmente espera porque otra sesión tiene un lock incompatible.

Terminar a la víctima no soluciona el bloqueador.

2. Suponer que la sesión con la consulta más larga es el bloqueador

La duración y los bloqueos son señales distintas.

Sigue la cadena de `blocking_session_id`.

3. Ignorar las sesiones inactivas

Una sesión inactiva puede seguir manteniendo locks a través de una transacción abierta.

Revisa siempre las sesiones y el estado de las transacciones cuando un bloqueador identificado no tenga una solicitud activa.

4. Tratar todo `blocking_session_id` distinto de cero como un SPID normal

SQL Server puede devolver valores negativos especiales en algunos escenarios de locks internos o distribuidos.

Usa los ID positivos para el análisis habitual de bloqueos entre sesiones.

5. Suponer que toda espera `LCK_M_*` significa «agregar un índice»

Un índice puede reducir la duración de los locks en algunas cargas de trabajo, pero los bloqueos también pueden deberse a:

- transacciones largas
- el alcance de las transacciones en la aplicación
- modificaciones por lotes grandes
- el comportamiento del nivel de aislamiento
- patrones de acceso en conflicto

Diagnostica la transacción y el plan de ejecución antes de cambiar índices.

6. Ignorar los límites de las transacciones

Una consulta puede terminar rápido mientras la transacción sigue abierta.

Pregúntate siempre si la aplicación confirma o revierte las transacciones con prontitud.

SIGUIENTE PASO

Usa el patrón de bloqueo para decidir qué revisar a continuación.

- **Una sesión bloquea a muchas otras** → investiga primero el bloqueador principal.
- **El bloqueador principal está activo** → revisa su SQL, su plan de ejecución, el alcance de su transacción, sus lecturas, escrituras y duración.
- **El bloqueador principal está inactivo** → revisa las transacciones abiertas y cómo maneja la aplicación el commit y el rollback.
- **Predominan las esperas** `LCK_M_*` → revisa el modo de lock, el recurso bloqueado, el tamaño de la transacción y el orden de acceso.
- **Los bloqueos provienen repetidamente de un mismo job o aplicación** → pasa de la respuesta a incidentes a la corrección de la aplicación o la consulta.
- **El bloqueo desaparece antes de capturarlo** → agrega Extended Events o una recopilación periódica de DMV.
- **No hay bloqueos** → vuelve al flujo más amplio de diagnóstico de rendimiento de SQL Server.

Para profundizar

- Guía rápida relacionada: Ver qué se está ejecutando en SQL Server
- Guía rápida relacionada: Inspeccionar una sesión de SQL Server
- Guía rápida relacionada: Encontrar transacciones abiertas
- Guía rápida relacionada: Tipos de espera de SQL Server
- Guía rápida relacionada: Flujo de diagnóstico de rendimiento de SQL Server
- Artículo relacionado: Cómo diagnosticar bloqueos en SQL Server
- Artículo relacionado: Cómo obtener el ID de sesión de SQL Server y ver qué está haciendo
- Artículo relacionado: SQL Server va lento: un flujo práctico de diagnóstico