Controlar la durabilidad de la transacción.

Las confirmaciones(commit) de transacciones de SQL Server pueden ser completamente duraderas, las predeterminadas de SQL Server o demoradas (también conocidas como confirmación diferida o Delayed Durability).

Las confirmaciones de transacciones totalmente duraderas son sincrónicas e informan una confirmación como exitosa y devuelven el control al cliente solo después de que los registros de registro de la transacción se escriben en el disco. Las confirmaciones de transacciones duraderas retrasadas son asíncronas e informan que una confirmación es exitosa antes de que los registros de registro de la transacción se escriban en el disco

Delayed Durability.

Se logra utilizando grabaciones de registros asíncronos en el disco. Los registros de transacciones se guardan en un búfer y se escriben en el disco cuando el búfer se llena o se produce un evento de lavado del búfer. La durabilidad de la transacción retrasada reduce la latencia y la contención dentro del sistema porque:

* El procesamiento de confirmación de transacción no espera a que finalice el registro de E / S y devuelva el control al cliente.
* Es menos probable que las transacciones concurrentes compitan por log IO; en cambio, el búfer de registro se puede vaciar al disco en fragmentos más grandes, lo que reduce la contención y aumenta el rendimiento.

**Cuando usar la durabilidad de la Delayed Durability.**

Algunos de los casos en los que podría beneficiarse del uso de la durabilidad de la transacción retrasada son:

**Puede tolerar alguna pérdida de datos.**  
Si puede tolerar cierta pérdida de datos, por ejemplo, donde los registros individuales no son críticos siempre que tenga la mayoría de los datos, entonces vale la pena considerar aplicarlo. Si no puede tolerar la pérdida de datos, no lo use.

**Está experimentando un cuello de botella en las escrituras del registro de transacciones.**  
Si sus problemas de rendimiento se deben a la latencia en las escrituras del registro de transacciones, es probable que su aplicación se beneficie del uso de la durabilidad de la transacción retrasada.

**Sus cargas de trabajo tienen una alta tasa de contención.**  
Si su sistema tiene cargas de trabajo con un alto nivel de contención, se pierde mucho tiempo esperando que se liberen los bloqueos. La durabilidad de la transacción retrasada reduce el tiempo de confirmación y, por lo tanto, libera bloqueos más rápido, lo que resulta en un mayor rendimiento.

**Configuración**

El DBA, puede controlar si los usuarios pueden usar la durabilidad de la transacción retrasada en una base de datos con la siguiente sentencia

ALTER DATABASE ... SET DELAYED\_DURABILITY = { DISABLED | ALLOWED | FORCED }

**Utilizacion**

### Control de nivel de bloque atómico: stored Procedures compilados de forma nativa

**DESHABILITADO**  
[predeterminado] Con esta configuración, todas las transacciones que se comprometen en la base de datos son totalmente duraderas, independientemente de la configuración del nivel de confirmación (DELAYED\_DURABILITY = [ON | OFF]).

**PERMITIDO**  
Con esta configuración, la durabilidad de cada transacción se determina a nivel de transacción: DELAYED\_DURABILITY = { OFF | EN }. Consulte [Control de nivel de bloque atómico: procedimientos almacenados compilados de forma nativa](https://docs.microsoft.com/en-us/sql/relational-databases/logs/control-transaction-durability?view=sql-server-ver15#CompiledProcControl) y [control de nivel COMMIT -Transact-SQL](https://docs.microsoft.com/en-us/sql/relational-databases/logs/control-transaction-durability?view=sql-server-ver15#bkmk_T-SQLControl) para obtener más información.

**FORZADO**  
Con esta configuración, cada transacción que se confirma en la base de datos se retrasa de forma duradera. Ya sea que la transacción especifique durabilidad completa (DELAYED\_DURABILITY = OFF) o no especifique, la transacción se demora. Esta configuración es útil cuando la durabilidad de la transacción retrasada es útil para una base de datos y no desea cambiar ningún código de aplicación.