

Tarea 13

1^{er} Autor

Moreno Santoyo Mariana

Universidad Nacional Autonoma de Mexico

CDMX, México

mariana.moreno.santoyo@gmail.com.com

Abstract—En el contexto de SQL Server, los niveles de aislamiento de transacciones determinan cómo se manejan los datos accesibles y modificables entre transacciones concurrentes, afectando la integridad de los datos y el rendimiento del sistema.

Index Terms—SQL Server, niveles de aislamiento, READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SNAPSHOT, SERIALIZABLE, concurrencia, integridad de datos, rendimiento, lecturas sucias, condiciones de carrera.

I. INTRODUCCIÓN

La documentación de Microsoft Learn aborda la configuración y los efectos de los niveles de aislamiento de transacciones en SQL Server, incluido cómo controlan el comportamiento de bloqueo y la versión de filas de las declaraciones Transact-SQL emitidas por una conexión. Los niveles de aislamiento descritos son READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SNAPSHOT y SERIALIZABLE, cada uno con un enfoque distinto en la visibilidad de los datos modificados por otras transacciones y en cómo se manejan los bloqueos y las versiones de filas. Se subraya la importancia de seleccionar el nivel adecuado para equilibrar entre el rendimiento y la precisión de los datos leídos durante las transacciones.

II. READ UNCOMMITTED

Especifica que las declaraciones pueden leer filas que han sido modificadas por otras transacciones pero que aún no se han confirmado.

Las transacciones configuradas con el nivel de aislamiento READ UNCOMMITTED no generan bloqueos compartidos, permitiendo así que otras transacciones alteren los datos que la transacción en curso está consultando. Del mismo modo, estas transacciones no se ven afectadas por bloqueos exclusivos que normalmente restringirían el acceso a filas modificadas, pero aún no confirmadas, por otras transacciones. Al emplear este nivel, se permite la lectura de modificaciones aún no confirmadas, conocidas como lecturas sucias, lo que significa que los valores de los datos pueden alterarse y las filas pueden surgir o desaparecer del conjunto de datos antes de que la transacción concluya. Utilizar este nivel de aislamiento equivale a aplicar NOLOCK a todas las tablas en cada sentencia SELECT dentro de una transacción, siendo este el nivel de aislamiento más permisivo.

III. READ COMMITTED

El nivel de aislamiento READ COMMITTED en SQL Server previene que las transacciones lean datos modificados pero no confirmados por otras transacciones, evitando las lecturas sucias. Aunque permite que los datos puedan cambiar entre lecturas consecutivas, lo que podría causar lecturas no repetibles o la aparición de datos fantasmas. SQL Server, por defecto, emplea bloqueos compartidos que restringen modificaciones durante una lectura y bloquean la lectura de filas modificadas hasta que la otra transacción concluye.

La implementación de READ COMMITTED varía según la configuración de READ_COMMITTED_SNAPSHOT:

- **Desactivada (predeterminado en SQL Server):** Se usan bloqueos compartidos que se liberan según el tipo de bloqueo (fila, página, tabla) conforme avanza la transacción.
- **Activada (predeterminado en Azure SQL Database):** Se utiliza el control de versiones de fila para proporcionar una vista instantánea de los datos al inicio de cada declaración, sin necesidad de bloqueos.

Cuando READ_COMMITTED_SNAPSHOT está activado, es posible optar por la sugerencia READCOMMITTEDLOCK para aplicar bloqueos compartidos en declaraciones específicas, permitiendo un control flexible entre el uso de bloqueos y el control de versiones de fila según la necesidad.

IV. REPEATABLE READ

En el nivel de aislamiento REPEATABLE READ, las transacciones están configuradas para no poder leer datos que han sido modificados pero aún no confirmados por otras transacciones. Además, previene que otras transacciones modifiquen los datos que han sido leídos por la transacción actual hasta que esta última se complete. Esto se logra a través del establecimiento de bloqueos compartidos sobre los datos leídos durante la transacción, los cuales se mantienen hasta el final de la misma.

Estos bloqueos compartidos impiden que otras transacciones modifiquen las filas leídas por la transacción en curso, aunque permiten la inserción de nuevas filas que satisfagan las condiciones de búsqueda de la transacción actual. Esto puede conducir a lo que se denomina “lecturas fantasma”, donde una declaración repetida en la transacción puede encontrar nuevas filas que antes no existían.

Dado que los bloqueos compartidos se mantienen hasta la conclusión de la transacción en lugar de liberarse tras

cada declaración, este nivel de aislamiento resulta en una simultaneidad inferior en comparación con el nivel `READ COMMITTED`, que es el predeterminado en muchas bases de datos. Por tanto, el uso del nivel `REPEATABLE READ` debería reservarse solo para aquellos casos en los que sea estrictamente necesario, debido a su impacto en la simultaneidad.

V. SNAPSHOT

El nivel de aislamiento `SNAPSHOT` asegura que cualquier declaración dentro de una transacción acceda a una versión de los datos que es transaccionalmente consistente con el estado de la base de datos al inicio de la transacción. En este nivel, una transacción solo es consciente de las modificaciones de datos que se confirmaron antes de su inicio, y no ve cambios realizados por otras transacciones después de ese momento. Esto crea un efecto de “instantánea” de los datos tal como existían al principio de la transacción.

Las transacciones `SNAPSHOT` no solicitan bloqueos al leer datos, permitiendo que otras transacciones modifiquen datos sin impedimentos. Sin embargo, durante una recuperación de base de datos, si una transacción `SNAPSHOT` intenta leer datos en conflicto con una transacción que se está revirtiendo, se bloqueará hasta que la reversión esté completa.

Para usar el nivel de aislamiento `SNAPSHOT`, la opción de base de datos `ALLOW_SNAPSHOT_ISOLATION` debe estar habilitada. Si una transacción `SNAPSHOT` accede a múltiples bases de datos, esta opción debe estar activada en todas ellas.

Es importante notar que una transacción no puede cambiar al nivel `SNAPSHOT` si ya comenzó con un nivel de aislamiento diferente. Una vez que una transacción utiliza `SNAPSHOT`, puede ver los cambios que ella misma ha realizado, como una actualización seguida por una selección en la misma tabla reflejará los cambios recientes.

VI. SERIALIZABLE

El nivel de aislamiento más restrictivo, conocido como `SERIALIZABLE`, especifica las siguientes reglas para las transacciones:

- Las declaraciones no pueden leer datos que hayan sido modificados pero aún no estén confirmados por otras transacciones.
- Ninguna otra transacción puede modificar los datos que la transacción actual ha leído hasta que esta última se complete.
- Otras transacciones no pueden insertar nuevas filas con valores clave que caigan dentro del rango de claves leídas por cualquier declaración en la transacción actual, hasta que esta se complete.
- Se colocan bloqueos de rango en los valores clave que coinciden con las condiciones de búsqueda de cada declaración ejecutada en la transacción, impidiendo que otras transacciones actualicen o inserten filas que cumplirían con los criterios de selección de la transacción actual. Esto asegura que, si se ejecuta nuevamente alguna declaración dentro de la transacción, leerá el mismo conjunto de filas.

Estos bloqueos de rango se mantienen hasta que la transacción se completa, lo que hace de `SERIALIZABLE` el nivel de aislamiento más restrictivo. Debido a que reduce la simultaneidad, se recomienda su uso solo cuando sea estrictamente necesario. Este nivel de aislamiento equivale a aplicar `HOLDLOCK` en todas las tablas de todas las declaraciones `SELECT` dentro de una transacción.

VII. ACID

Las transacciones en el contexto de bases de datos y sistemas de gestión de transacciones están regidas por el acrónimo `ACID`, que representa las cuatro propiedades fundamentales de una transacción: Atomicidad, Coherencia, Aislamiento y Durabilidad.

- **Atomicidad** Esta propiedad asegura que todos los cambios en los datos ocurran como si fueran una única operación. Por lo tanto, o bien se realizan todos los cambios requeridos por la transacción, o ninguno de ellos. Por ejemplo, en una transferencia de fondos de una cuenta a otra, la atomicidad garantiza que si se debita una cuenta, la otra cuenta será acreditada correspondientemente.
- **Coherencia** Garantiza que cada transacción lleva a la base de datos de un estado coherente a otro. Esto significa que los datos deben ser consistentes antes y después de la transacción. En el contexto de la transferencia de fondos, asegura que el total de fondos antes y después de la transacción permanezca igual.
- **Aislamiento** Esta propiedad asegura que las operaciones intermedias de una transacción no son visibles para otras transacciones. Esto hace que las transacciones concurrentes parezcan ser serializadas. En el caso de la transferencia de fondos, el aislamiento previene que una transacción vea los fondos en un estado intermedio.
- **Durabilidad** Asegura que una vez que una transacción ha sido completada, los cambios que ha realizado persistirán, incluso en caso de fallas del sistema. Esto significa que, siguiendo con el ejemplo de la transferencia de fondos, una vez completada, los cambios en las cuentas involucradas son definitivos y no se pueden revertir.

Estas propiedades juntas garantizan la confiabilidad y la consistencia de las transacciones en sistemas de bases de datos, permitiendo así la gestión segura y efectiva de los datos.

REFERENCES

- WilliamDAssafMSFT, “SET TRANSACTION ISOLATION LEVEL (Transact-SQL) - SQL Server.” <https://learn.microsoft.com/en-us/sql/t-sql/statements/set-transaction-isolation-level-transact-sql?view=sql-server-ver16>
- “IBM documentation,” Dec. 22, 2022. <https://www.ibm.com/docs/es/cics-tx/11.1?topic=processing-acid-properties-transactions>
- rwestMSFT, “Transaction Isolation Levels - SQL Server,” Microsoft Learn, Feb. 28, 2023.

<https://learn.microsoft.com/en-us/sql/t-sql/language-elements/transaction-isolation-levels?view=sql-server-ver16>