

Práctica A: Administración de transacciones y bloqueos

Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Definir transacciones con las instrucciones `BEGIN TRANSACTION` y `COMMIT TRANSACTION`.
- Determinar el número de transacciones activas al consultar la variable global `@@trancount`.
- Utilizar el procedimiento almacenado de sistema `sp_lock` y el Administrador corporativo de SQL Server para ver la información de los bloqueos.
- Utilizar la instrucción `SET TRANSACTION ISOLATION LEVEL` para controlar el comportamiento de los bloqueos en el nivel de sesión.
- Utilizar las opciones de bloqueo de tabla para controlar el comportamiento de los bloqueos en tablas específicas.
- Utilizar la instrucción `SET LOCK_TIMEOUT` para controlar la cantidad máxima de tiempo que una instrucción puede esperar a que se libere un bloqueo.

Requisitos previos

- El archivo `lab03.zip` contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.
- Para la realización de las prácticas es necesario ejecutar el script `Restore.cmd` del directorio `Restore`.

Ejercicio 1

Creación y ejecución de una transacción

En este ejercicio, va a utilizar las instrucciones `BEGIN TRANSACTION` y `COMMIT TRANSACTION` para comprender el efecto que surten las instrucciones en la forma en la que se modifican los datos. También va a ver cómo SQL Server utiliza la variable global `@@trancount` para determinar si una transacción está activa.

✍ Para crear y ejecutar una transacción

En este procedimiento, va a utilizar las instrucciones `BEGIN TRANSACTION` y `COMMIT TRANSACTION` para controlar cómo se procesa una instrucción `UPDATE` sobre la tabla **Customers**.

1. Abra el Analizador de consultas SQL y, si así se le pide, conéctese al servidor local con autenticación de Microsoft Windows.
2. En la lista **BD**, haga clic en **ClassNorthwind**.
3. Abra el archivo Tran1.sql y revise su contenido.

Observe que la instrucción `BEGIN TRAN` está seguida por la instrucción `UPDATE`, pero no se encuentra la instrucción `COMMIT TRAN` o `ROLLBACK TRAN` correspondiente. Las instrucciones `SELECT` y `PRINT`, y la variable global `@@trancount` se utilizan en la secuencia de comandos para mostrar el progreso de la transacción.

4. Ejecute la secuencia de comandos y revise los resultados.

En este momento, ¿se han confirmado los cambios efectuados con la instrucción UPDATE en esta transacción? ¿Cómo puede determinar esto?

No, la transacción se debe completar con la instrucción COMMIT TRAN. La transacción sigue estando activa y sigue manteniendo los bloqueos que haya adquirido. El valor de la variable global @@trancount es 1, lo que indica que en esta sesión se ha enviado una instrucción BEGIN TRAN.

¿Podrían otras transacciones consultar o actualizar los datos modificados?

Otras transacciones no podrían consultar o modificar los datos modificados hasta que se haya confirmado (o deshecho) la transacción.

5. Escriba una instrucción COMMIT TRANSACTION en la ventana de consulta y, después, resáltela y ejecútela para completar la transacción y hacer que los cambios sean definitivos.
6. Resalte y ejecute una de las instrucciones SELECT de la tabla **Customers** para comprobar que ahora sí se ha completado el cambio.

Ejercicio 2

Deshacer una transacción

En este ejercicio, va a utilizar la instrucción `ROLLBACK TRANSACTION` para comprender el efecto que tiene la forma en la que se modifican los datos dentro de una transacción.

✍ Para utilizar la instrucción `ROLLBACK TRANSACTION`

En este procedimiento utilizará las instrucciones `BEGIN TRANSACTION` y `ROLLBACK TRANSACTION` para controlar el modo en que se procesa una instrucción `UPDATE` en la tabla **member**.

1. Abra el archivo `Tran2.sql` y revise su contenido.

Observe que este archivo de comandos es similar a `Tran1.sql`, pero el nombre de contacto es distinto y se ha agregado una instrucción `ROLLBACK TRAN`.

2. Ejecute la secuencia de comandos y revise los resultados.

¿Se ha almacenado de forma permanente en la base de datos la modificación efectuada por la instrucción `UPDATE`?

No. La transacción se ha cancelado, de modo que los cambios efectuados durante la transacción se han deshecho.

¿Se ha completado la transacción?

Sí. La instrucción `ROLLBACK TRAN` completa la transacción y libera los bloqueos adquiridos por la transacción. La consulta de la variable global `@@trancount` devuelve cero.

Ejercicio 3

Presentación de la información de bloqueo

En este ejercicio va a ejecutar simultáneamente varias transacciones para determinar el efecto de dicha actividad sobre los bloqueos. Se le pedirá que abra varias conexiones con el Analizador de consultas SQL para simular que varios usuarios están enviando transacciones a SQL Server.

Observe que las secuencias de comandos de este ejercicio no siempre incluyen instrucciones COMMIT TRAN o ROLLBACK TRAN. Su ausencia mantiene las transacciones abiertas y los bloqueos asociados activos, para poder ver así la información de bloqueo.

✍ Para ver la información de bloqueo

En este procedimiento va a utilizar las instrucciones BEGIN TRANSACTION y ROLLBACK TRANSACTION para controlar cómo se procesa una instrucción UPDATE sobre la tabla **Customers**.

1. Inicie el Analizador de consultas SQL (conexión 1), y haga clic en **Borrar la ventana de Consulta**.
2. Ejecute el procedimiento almacenado de sistema **sp_lock** y revise el resultado.
3. Inicie el Administrador corporativo de SQL Server.
4. En el Administrador corporativo de SQL Server, en el árbol de la consola, expanda su servidor, expanda **Administración** y, después, expanda **Actividad actual**. Revise la información que se presenta en **Información del proceso**, **Bloqueos / Id. de proceso** y **Bloqueos / Objeto**.
5. Abra una segunda conexión en el Analizador de consultas SQL (conexión 2) y, en la lista **BD**, haga clic en **ClassNorthwind**.
6. Abra el archivo Lock1.sql mediante la conexión 2 y revise su contenido.
Observe que se inicia una transacción con la instrucción BEGIN TRAN pero no existe la correspondiente instrucción COMMIT TRAN o ROLLBACK TRAN para completar la transacción.
7. Ejecute Lock1.sql mediante la conexión 2 y revise los resultados.
8. Cambie a la conexión 1, ejecute el procedimiento almacenado de sistema **sp_lock** y revise la información de bloqueo.

Identifique los diferentes tipos de bloqueo y los recursos bloqueados por la transacción. Anote dicha información para utilizarla en el Ejercicio 4.

9. Cambie al Administrador corporativo de SQL Server, haga clic con el botón secundario del *mouse* (ratón) en **Actividad actual** y haga clic en **Actualizar**.
10. Revise la información de **Actividad actual** que se presenta en **Información del proceso, Bloqueos / Id. de proceso** y **Bloqueos / Objeto**.
11. Cambie a la conexión 2 y cancele la transacción iniciada en el paso 7; para ello, ejecute una instrucción ROLLBACK TRAN.
12. Cambie a la conexión 1 y ejecute el procedimiento almacenado de sistema **sp_lock**.

Observará que los bloqueos adquiridos por la transacción en el paso 6 quedan ahora liberados.

Ejercicio 4

Establecimiento de las opciones de bloqueo

En este ejercicio, va a utilizar algunas opciones de bloqueo de SQL Server para determinar cómo afectan a la forma en la que se procesan las transacciones. Va a utilizar las conexiones que estableció en el Analizador de consultas SQL en el ejercicio 3 para simular que varios usuarios están enviando transacciones a SQL Server. También puede utilizar la ventana Actividad actual del Administrador corporativo de SQL Server para ver la información de bloqueo durante este ejercicio.

✍ Para establecer el nivel de aislamiento de la transacción

En este procedimiento va a utilizar la instrucción SET TRANSACTION ISOLATION LEVEL para controlar el comportamiento del bloqueo en el nivel de sesión. Va a utilizar las conexiones que estableció en el Analizador de consultas SQL en el ejercicio 3.

1. Cambie al Analizador de consultas SQL (conexión 2).
2. Abra el archivo de comandos Lock2.sql mediante la conexión 2, revise su contenido y, después, ejecútelo.
3. Cambie a la conexión 1, ejecute el procedimiento almacenado de sistema **sp_lock** y revise la información de bloqueo.

Identifique y revise los diferentes tipos de bloqueo y los recursos que la transacción ha bloqueado.

¿Ha utilizado SQL Server bloqueos diferentes de los que ha utilizado en el paso 8 del ejercicio 3? Razone la respuesta.

Esta vez, SQL Server ha utilizado un bloqueo de tabla exclusivo porque la instrucción SET TRANSACTION ISOLATION LEVEL indicó que se utilizara el tipo de bloqueo que se requiere para conseguir el nivel de aislamiento especificado.

-
4. Cambie a la conexión 2 y cancele la transacción; para ello, ejecute una instrucción ROLLBACK TRAN.

✍ Para utilizar las opciones de bloqueo de tablas

En este procedimiento va a utilizar opciones de bloqueo de tabla para controlar el comportamiento de los bloqueos.

1. Cambie al Analizador de consultas SQL (conexión 1).
2. Ejecute el procedimiento almacenado de sistema **sp_lock** mediante la conexión 1 y revise el resultado.
3. Cambie al Analizador de consultas SQL (conexión 2).

4. Abra el archivo de comandos Lock3.sql mediante la conexión 2 y revise su contenido.

Observe que se ha definido una opción de bloqueo de tabla en la cláusula FROM de la instrucción SELECT. Observe además que la secuencia de comandos no contiene una instrucción COMMIT TRAN o ROLLBACK TRAN.

5. Ejecute el archivo de comandos Lock3.sql en la conexión 2.
6. Cambie a la conexión 1 y ejecute el procedimiento almacenado de sistema **sp_lock**.

Anote los tipos de bloqueo que se utilizan y los recursos que están bloqueados.

7. Abra una tercera conexión en el Analizador de consultas SQL (conexión 3) y, en la lista **BD**, haga clic en **ClassNorthwind**.
8. Abra el archivo de comandos Lock1.sql mediante la conexión 3 y ejecútelo.
9. Cambie a la conexión 1 y ejecute el procedimiento almacenado de sistema **sp_lock**.

¿Hay una transacción que no se puede ejecutar? Si es así, ¿por qué?

Sí, la segunda transacción está intentando actualizar los datos que están bloqueados por la primera transacción, de modo que la segunda transacción debe esperar a que termine la primera. Puede determinar si una transacción no se puede ejecutar debido a un conflicto de bloqueo; para ello, utilice el procedimiento almacenado de sistema sp_lock y busque la palabra WAIT en la columna de estado. En este caso, la transacción de la conexión 3 debe esperar a que termine la transacción de la conexión 2.

¿Cuánto tiempo espera una transacción un recurso bloqueado?

Una transacción espera indefinidamente un recurso bloqueado, siempre y cuando no haya interbloqueos. Se puede establecer el período de tiempo de espera de la sesión para controlar cuánto tiempo esperará SQL Server a que se liberen los recursos bloqueados.

10. Cambie a la conexión 3 y, en la barra de herramientas, haga clic en **Cancelar ejecución de consulta**.
11. Cambie a la conexión 1 y ejecute el procedimiento almacenado del sistema **sp_lock** para comprobar que la transacción en espera se ha cancelado.

✍ Para establecer el período de tiempo de espera de bloqueo de una transacción

En este procedimiento, va a establecer el período de tiempo de espera de bloqueo para que la transacción espere la obtención de un bloqueo durante una cantidad de tiempo especificada.

1. Cambie a la conexión 3, modifique el archivo de comandos Lock1.sql y agregue la siguiente instrucción inmediatamente antes de la instrucción BEGIN TRAN:

```
SET lock_timeout 500
```

2. Ejecute la secuencia de comandos modificada mediante la conexión 3.

¿Qué ocurre y por qué?

La transacción de la conexión 3 estaba esperando los recursos bloqueados por la transacción de la conexión 2. SQL Server canceló la transacción de la conexión 3 porque terminó el tiempo de espera de bloqueo que se especificó.

3. Cierre todas las ventanas del Analizador de consultas SQL.