



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

Base de Datos

Tarea 22

Profesor: Ing. Fernando Arreola
Franco

Grupo: 01

Alumna: Montes Cantero Zelene
Yosseline Isayana.

Semestre 2022-2

Fecha de entrega: 29/Abril/2022



Niveles de aislamiento

Una transacción es una unidad atómica de trabajo que se completa en su totalidad o no se lleva a cabo en absoluto. Para fines de recuperación, el sistema debe hacer el seguimiento de cuándo se inicia, termina y confirma o aborta una transacción. Por consiguiente, el gestor de recuperación hace un seguimiento de las siguientes operaciones:

- **BEGIN_TRANSACTION.** Marca el inicio de la ejecución de una transacción.
- **READ o WRITE.** Especifican operaciones de lectura o escritura en los elementos de la base de datos que se ejecutan como parte de una transacción.
- **END_TRANSACTION.** Especifica que las operaciones READ y WRITE de la transacción han terminado y marca el final de la ejecución de la transacción. Sin embargo, en este punto puede ser necesario comprobar si los cambios introducidos por la transacción pueden aplicarse de forma permanente a la base de datos (confirmados) o si la transacción se ha cancelado porque viola la serialización o por alguna otra razón.
- **COMMIT_TRANSACTION.** Señala una finalización satisfactoria de la transacción, por lo que los cambios (actualizaciones) ejecutados por la transacción se pueden enviar con seguridad a la base de datos y no se desharán.
- **ROLLBACK (o ABORT).** Señala que la transacción no ha terminado satisfactoriamente, por lo que deben deshacerse los cambios o efectos que la transacción pudiera haber aplicado a la base de datos.

Los niveles de aislamiento de transacción se definen por la presencia o ausencia de los siguientes fenómenos:

•*Lecturas desdesociadas*

Una lectura desensaciada se produce cuando una transacción lee datos que aún no se han confirmado. Por ejemplo, suponga que la transacción 1 actualiza una fila. La transacción 2 lee la fila actualizada antes de que la transacción 1 confirme la actualización. Si la transacción 1 revierte el cambio, la transacción 2 tendrá datos de lectura que se considera que nunca han existido.

•*Lecturas no irrepetibles*

Una lectura no irrepetible se produce cuando una transacción lee la misma fila dos veces, pero obtiene datos diferentes cada vez. Por ejemplo, suponga que la transacción 1 lee una fila. La transacción 2 actualiza o elimina esa fila y confirma la actualización o eliminación. Si la transacción 1 vuelve a leer la fila, recupera distintos valores de fila o detecta que se ha eliminado la fila.

•*Fantasmas*

Un fantasma es una fila que coincide con los criterios de búsqueda, pero que no se ve inicialmente. Por ejemplo, supongamos que la transacción 1 lee un conjunto de filas que cumplen algunos criterios de búsqueda. La transacción 2 genera una nueva fila (mediante una actualización o una inserción) que coincide con los criterios de búsqueda de la transacción 1. Si la transacción 1 vuelve a ejecutar la instrucción que lee las filas, obtiene un conjunto diferente de filas.

Los cuatro niveles de aislamiento de transacción (como se define SQL-92) se definen en términos de estos fenómenos. En la tabla siguiente, una "X" marca cada fenómeno que puede producirse.

Nivel de aislamiento de transacción	Lecturas desde so	Lecturas no irrepetibles	Fantasmas
Lectura no confirmada	X	X	X
Lectura confirmada	--	X	X
Lectura repetible	--	--	X
Serializable	--	--	--

En la tabla siguiente se describen formas sencillas de que un DBMS pueda implementar los niveles de aislamiento de transacción

Aislamiento de transacciones		Posible implementación
Lectura no confirmada		Las transacciones no están aisladas entre sí. Si dbms admite otros niveles de aislamiento de transacción, omite el mecanismo que usa para implementar esos niveles. Para que no afecten negativamente a otras transacciones, las transacciones que se ejecutan en el nivel Read Uncommitted suelen ser de solo lectura.
Lectura confirmada		<p>La transacción espera hasta que se desbloquean las filas bloqueadas por escritura por otras transacciones; esto impide que lea los datos "desnuciado".</p> <p>La transacción contiene un bloqueo de lectura (si solo lee la fila) o un bloqueo de escritura (si actualiza o elimina la fila) en la fila actual para evitar que otras transacciones lo actualicen o eliminen. La transacción libera bloqueos de lectura cuando se mueve fuera de la fila actual. Mantiene bloqueos de escritura hasta que se confirma o se revierte.</p>
Lectura repetible		<p>La transacción espera hasta que se desbloquean las filas bloqueadas por escritura por otras transacciones; esto impide que lea los datos "desnuciado".</p> <p>La transacción contiene bloqueos de lectura en todas las filas que devuelve a la aplicación y escribe bloqueos en todas las filas que inserta, actualiza o elimina. Por ejemplo, si la transacción incluye SQL instrucción SELECT * FROM Orders, la transacción lee las filas a medida que la aplicación las captura. Si la transacción incluye SQL instrucción DELETE FROM Orders WHERE Status = 'CLOSED', la transacción bloquea las filas a medida que las elimina.</p> <p>Dado que otras transacciones no pueden actualizar ni eliminar estas filas, la transacción actual evita las lecturas no irrepetibles. La transacción libera sus bloqueos cuando se confirma o se revierte.</p>

La transacción espera hasta que se desbloquean las filas bloqueadas por escritura por otras transacciones; esto impide que lea los datos "desnuciado".

Serializable

La transacción contiene un bloqueo de lectura (si solo lee filas) o un bloqueo de escritura (si puede actualizar o eliminar filas) en el intervalo de filas a los que afecta. Por ejemplo, si la transacción incluye la instrucción SQL **SELECT * FROM Orders**, el intervalo es toda la tabla Orders; la transacción bloquea la tabla y no permite insertar nuevas filas en ella. Si la transacción incluye SQL instrucción **DELETE FROM Orders WHERE Status = 'CLOSED'**, el intervalo es todas las filas con un estado de "CLOSED"; la transacción bloquea todas las filas de la tabla Orders con el estado "CLOSED" y no permite insertar ni actualizar ninguna fila de forma que la fila resultante tenga el estado "CLOSED".

Dado que otras transacciones no pueden actualizar ni eliminar las filas del intervalo, la transacción actual evita las lecturas no irrepetibles. Dado que otras transacciones no pueden insertar ninguna fila en el intervalo, la transacción actual evita los fantasmas. La transacción libera su bloqueo cuando se confirma o se revierte.

Bibliografía

- *Niveles de aislamiento de transacciones (ODBC) - ODBC API Reference.* (s. f.). Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/es-es/sql/odbc/reference/develop-app/transaction-isolation-levels?view=sql-server-ver15>
- Ing. Czeslaw Kristofer Lugowski Rivero. Apuntes de clase "Laboratorio de base de datos". Semestre 2022-2. Facultad de Ingeniería UNAM.