|  |  |  |
| --- | --- | --- |
|  | | |
|  | **Hurlingham**, Agosto del 2021  Buenos Aires, Argentina | |
| **ESTRATEGIAS DE PERSISTENCIA (759\_45)**  Profesor: Lic. Pablo Marcelli  **TP\_CLASE2\_TEORICA**  Alumno: Raul A. Gutierrez  correo electrónico: [correo@raulgutierrez.com.ar](mailto:correo@raulgutierrez.com.ar)  [raul.gutierrez@estudiantes.unahur.edu.ar](mailto:raul.gutierrez@estudiantes.unahur.edu.ar)  Entrega: Lunes 23 de Agosto 2021 | |  |

1.- ¿Qué es una transacción y que propiedades tiene explíquelas?

2.- ¿Cuáles son los estados de una transacción?

3.- ¿A qué se refiere el término, control de concurrencia?

4.- ¿Qué es el bloqueo en modo compartido y en modo exclusivo?

5.- ¿Qué son los protocolos de bloqueos en dos fases?

6.- ¿Cómo es la recuperación de un interbloqueo?

7.- ¿A qué hace referencia, los niveles débiles de consistencia?

1.- ¿Qué es una transacción y que propiedades tiene explíquelas?

Una transacción, es la unidad de ejecución de un programa, que accede, consulta y posiblemente, actualice varios elementos de datos.

Una transacción, se inicia por la ejecución de un programa que la contiene, y está delimitada por las instrucciones que existen entre el inicio de la transacción (**begin transaction**) y el fin de la transacción (**end transaction**). En definitiva, una transacción consiste en el conjunto de todas las operaciones que se ejecutan entre estos dos eventos.

Propiedades de las transacciones

* Atomicidad: dictamina que, o todas las transacciones se realizan de forma adecuada sobre la base de datos, o, ninguna de ellas surte efecto sobre los datos de la base.

La responsabilidad de mantener la atomicidad es del sistema de base de datos.

* Consistencia: establece que toda ejecución sea aislada para toda transacción, conservando la inalterabilidad (consistencia) de los datos durante su ejecución. Esto es para que las transacciones sean únicas y no exista forma de que se ejecute otra transacción de manera concurrente, a la que se está ejecutando.

La responsabilidad de mantener este estado es del programador, asegurando la consistencia del manejo de los datos en la aplicación que codifica dicha transacción.

* Aislamiento: Aunque se ejecuten varias transacciones concurrentemente, el sistema garantiza que cada transacción ignora al resto, mientras se esté ejecutando. Asegurando de esta forma que los cambios que pueda realizar cada transacción, no se producirán antes de que finalice la ejecución de la anterior. La responsabilidad de asegurar la propiedad de aislamiento, es de un componente del sistema de base de datos, llamado control de concurrencias.
* Durabilidad: tras la finalización con éxito de una transacción, los cambios realizados por esta en la base son permanentes, incluso si hay fallos en el sistema. La responsabilidad de asegurar la durabilidad, pertenece a un componente de software en el sistema de base de datos, llamado componente de gestión de recuperaciones.

2.- ¿Cuáles son los estados de una transacción?

En la ausencia de fallos, todas las transacciones se completan con éxito. Entonces, los estados de una transacción, son como fue a partir del inicio, la terminación y el éxito (o no) que tuvo la ejecución de la misma.

Se establece entonces un simple modelo de abstracción, ya que toda transacción puede tener cualquiera de los siguientes estados:

* Activa: es el estado inicial, en el que permanece la transacción durante su ejecución.
* Fallida: es el estado de la transacción, tras descubrir que no puede continuar la ejecución normal. La falla termina la ejecución, y la transacción pasa al estado de abortada.
* Abortada: es la transacción que no termina con éxito su ejecución, por ende, no debe tener efecto sobre el estado de los datos y cualquier cambio efectuado, se deshacen.
* Retrocedida: al abortarse una transacción, los cambios realizados por esta se revierten, a esta transacción ahora se la reconoce como retrocedida.
* Parcialmente comprometida: es el momento justo antes de ejecutarse la última instrucción de la transacción, y de no haber error después pasara al estado de comprometida.
* Comprometida: Es la transacción que termina con éxito, y en caso de haber hecho cambio en el estado de los datos, sus cambio ya no pueden ser revertidos. Ni siquiera abortando la transacción. Tras completarse con éxito, los cambios que ya se han hecho y transformado los datos de la base, la han llevado a un nuevo estado consistente y permanente. Incluso aunque haya un fallo en el sistema. Cuando una transacción se ha comprometido, no se pueden deshacer sus efectos. La única forma es ejecutando una transacción compensadora.
* Compensadora: si una transacción comprometida, realizo un incremento inesperado (por ejemplo), es necesario realizar otra transacción que compense ese incremento, realizando una quita. Sin embargo no siempre se puede crear dichas transacciones, por lo tanto se deja la responsabilidad de crear y ejecutar dichas compensaciones al usuario, y no es parte de la gestión que pueda hacer el motor de la base de datos.

3.- ¿A qué se refiere el término, control de concurrencia?

Cuando una transacción accede a un elemento de datos, ninguna otra transacción puede modificar dicho elemento. Es una forma de asegurar la secuencialidad, exigir que el acceso a elementos se haga por exclusión mutua.

El método más habitual que se utiliza para implementar este requisito es permitir que una transacción acceda a un elemento de datos solo si posee el bloqueo de dicho elemento.

Existe un componente en el sistema de base de datos que se llama control de concurrencia. Este puede asegurar la consistencia de la base de datos en una ejecución concurrente, solo si está seguro de que cualquier planificación que ejecute transacciones de forma concurrente, tiene los mismos efectos de otra que se hubiese ejecutado sin concurrencia (ósea, en planificación secuencial).

4.- ¿Qué es el bloqueo en modo compartido y en modo exclusivo?

Modo compartido: es cuando una transacción tiene un bloqueo compartido sobre un elemento, al cual puede acceder para leer, pero no para escribir.

Modo exclusivo: es cuando la misma transacción tiene acceso al elemento, pero esta esta vez también puede escribir. Este es un modo exclusivo.

Esto dependerá desde ya de la solicitud que se haga al gestor de control de concurrencias, sobre qué tiene de bloqueo se necesita tener sobre un elemento determinado.

5.- ¿Qué son los protocolos de bloqueos en dos fases?

Este protocolo exige que cada transacción realice las peticiones de bloqueo y desbloqueo de dos fases:

- Una transacción puede obtener bloqueos pero no liberarlo. Esta es la fase de crecimiento. Inicialmente una transacción está en esta fase. La transacción adquiere los bloqueos que necesita.

- Una transacción puede liberar bloqueos, pero no puede liberarlos. Esta es la fase de decrecimiento. Una vez que la transacción libera un bloqueo, entra en la fase de decrecimiento y no puede realizar más solicitudes de bloqueo.

6.- ¿Cómo es la recuperación de un interbloqueo?

Cuando un algoritmo de detección de interbloqueos determina que existe un interbloqueo, el

Sistema debe recuperarse del mismo. La opción más común es retroceder las transacciones necesarias para romper el interbloqueo.

Para esto, se realizan tres acciones:

1.- selección de una víctima. Dado un conjuntos de transacciones interbloque das, se deberá determinar la transacción o transacciones, que se van a retroceder para romper el interbloqueo. El retroceso debe estar marcado por un coste mínimo de transacciones. Desafortunadamente, el determinar el coste mínimo no es preciso.

2.- Retroceso: una vez que se ha decido retroceder una transacción en particular, se debe determinar hasta donde retrocederá dicha transacción.

3.- Inanición: En un sistema en el cual la selección de víctimas está basada principalmente en factores de coste, puede ocurrir que siempre se elija a la misma transacción. El resultado es que esta transacción no completa nunca su tarea designada. Dicha situación se denomina inanición. Se debe asegurar que una transacción pueda ser elegida como víctima solo un número finito de veces.

7.- ¿A qué hace referencia, los niveles débiles de consistencia?

Si todas las transacciones tienen la posibilidad de mantener la consistencia de la base de datos si se ejecutan por separado o en la secuencialidad, asegura que las ejecuciones concurrentes mantengan la consistencia es tarea del control de concurrencias. Sin embargo, puede que los protocolos necesarios para asegurar la secuencialidad permitan muy poca concurrencia para algunas aplicaciones.

En estos casos se utilizan niveles más débiles de consistencia, y el uso de estos niveles de añade una nueva responsabilidad sobre los programadores, para asegurar la corrección y consistencia de las bases de datos.