Universidad del Valle de Guatemala
Bases de Datos

Davis Alvarez
15842

Laboratorio 10

Ejercicio 2

Para cada una de las funciones del ejercicio anterior discuta los posibles problemas de atomicidad (si los hubiera) que podrían ocurrir si el sistema fuera interrumpido durante cualquier punto de la ejecución de la función. Su respuesta debe incluir el análisis para cada una de las 4 funciones.

a). Buscar por Velocidad y RAM

El problema de atomicidad en esta función puede apareces justo en el momento de ejecutar el select, ya que, si obtenemos los datos y al mismo tiempo son cambiados por otra transacción, los datos obtenidos no serán los reales.

b). Eliminar por Modelo

En este caso el problema de atomicidad no se da, ya que solo nos interesa eliminar de las tablas, y con ayuda del commit, nos aseguramos de que se borren los dos registros de las tablas o no se borre ningún registro de las tablas.

c). Decrecer el precio por Modelo

Para este inciso el problema de atomicidad radica en que, si el cliente se tarda mucho tiempo en confirmar la transacción, otra transacción puede modificar o eliminar cualquier parte del registro y cuando el cliente decida confirmar la transacción el resultado ya no será el que el esperaba.

d). Agregar Modelo a la DB

El problema de atomicidad en esta función puede aparecer si el cliente se tarda mucho tiempo en confirmar la transacción, ya que otra transacción puede modificar las tablas de la DB con información parecida y al confirmar la transacción puede llegar a ejecutarse solamente parte de la transacción.

Ejercicio 3

Suponga que se ejecuta como una transacción T una de las cuatro funciones del ejercicio 1, mientras otras transacciones de ese mismo o de cualquier otro de los programas ocurren casi al mismo tiempo. ¿Qué diferencias podría observar para el resultado de T si todas las transacciones se ejecutan con un nivel de aislamiento READ UNCOMMITED en lugar de SERIALIZABLE?. Considere por separado el caso en que T es cada una de las 4 funciones definidas, es decir que su respuesta debe incluir el análisis para cada una de las 4 funciones

a). Buscar por Velocidad y RAM

En este caso el problema que se puede llegar a dar es que se haga una “lectura basura”, ya que alguna otra T pudo modificar el contenido de la tabla, pero no se ha hecho el commit y los valores van a cambiar. Por lo que los datos leídos no son los correctos hasta que se realice el commit y se vuelva a leer de nuevo.

b). Eliminar por Modelo

En este caso el inconveniente que puede existir si alguna T modifica el atributo por el que se pensaba borrar, en este caso, cuando se haga el commit cambiara el registro y al ejecutarse el delete no se encontrara y no se podrá borrar.

c). Decrecer el precio por Modelo

Para este inciso el problema es el de cualquier update que este esperando el commit y que modifique exactamente el registro que se quiere modificar con esta función. El resultado final, no seria el que el usuario esperaba, ya que se modifico antes de hacer la transacción y al registro ya modificado se le aplica los cambios que el usuario quería ir.

d). Agregar Modelo a la DB

El problema de utilizar READ UNCOMMITED en este caso es que si alguna otra T realizó un insert a la tabla con la misma llave primario que va a realizar esta función, va a lanzar un error por tratar de ingresar registros con la misma llave.