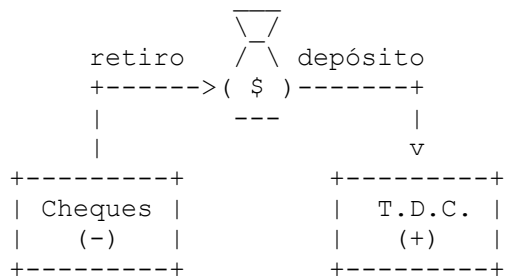


**TRANSACCIONES.**

Una transacción es un conjunto de operaciones que deben realizarse todas o ninguna. Un ejemplo claro de esto es una transferencia bancaria, por ejemplo un pago a tarjeta de crédito:



Esta operación requiere de dos pasos: un retiro y un depósito, para que dicha operación se considere exitosa deben realizarse ambas, de lo contrario ninguna debe ser aplicada, este ejemplo puede antojarse simplista, pero imagina miles de operaciones de este tipo en línea, bien por sistema o bien por internet. Si alguna operación de este tipo quedara a medias y en firme, el resultado sería una base de datos inconsistente, ya que la información almacenada en la BD no correspondería con la realidad (una cuenta de la que se sacaron \$1,000.00 pero que nunca se depositaron en la segunda cuando en la realidad la operación se llevó a cabo)

Para este fin necesitamos de un medio que nos permita mantener registro de cada actividad perteneciente a cada transacción en el ambiente de base datos, este espacio destinado a guardar las transacciones le llamamos bitácora de transacciones o logical log.

Esta bitácora es usada para cuatro cosas:

- Dar marcha atrás (rollback) a la transacción en caso de errores o por código.
- Asegurar la integridad relacional de la base de datos en caso de caída total del RDBMS o del equipo (lo veremos en un capítulo de integridad de datos).
- Dado que las transacciones guardan todos los SQL que tienen que ver con actualización (UPDATE, INSERT o DELETE) pueden respaldarse a fin de utilizarlos para llevar a un punto en el tiempo más reciente en caso de tener que hacer un restore de un respaldo (lo veremos en mecanismos de disponibilidad).
- Para fines de replicación de datos basadas en transacciones (lo veremos en mecanismos de disponibilidad).

La bitácora de transacciones, de ahora en adelante logical logs, reside en un espacio en disco finito y determinado, aunque es posible que se guarden en archivos externos a nivel de S.O. lo más común es que sean administrados en una porción de uno o varios chunks del RDBMS y rara vez, por no decir nunca, radica en un solo archivo, o logical log file, de hecho, es raro y muy poco recomendable, el tener menos de tres logical logs files.



Cabe mencionar que cuando a una transacción se le da marcha atrás, los pasos necesarios para regresar al principio de esta, (por ejemplo para un INSERT en la transacción necesitamos aplicar un DELETE), son almacenados también en la bitácora de transacciones y reciben el nombre de registros de compensación.

Los logical log files son unidades discretas y que se liberan cuando:

- Todas las transacciones que tienen operaciones en dicho logical log file han terminado exitosamente (COMMIT) o han terminado de dar marcha atrás (ROLLBACK) y
- Haya sido respaldadas en caso de que se haya configurado el uso de respaldo de logical logs.

Estos logical logs files se usan de forma circular, es decir, se comienza en el logical log file 1, una vez lleno se usa el dos, que a su vez al llenarse nos fuerza a usar un tercero, si éste es el último y el primero ha sido liberado, procederemos a usarlo nuevamente y así sucesivamente.

		_
		v
Logical Log		
File 1		v
Logical Log		
File 1		v
:		
:		v
Logical Log		
File n		_

Sin embargo ¿qué sucedería si el tercer logical log file se llena y el primero aún no ha sido liberado? En esta situación, logical log files llenos, la bitácora ya no tendría más espacio para guardar las transacciones entrantes, peor aún, ni siquiera tendría la posibilidad de abortar las transacciones presentes por que no podría almacenar los registros de compensación. Esto puede llevarnos a un estado de inactividad total, más aún, algunos manejadores de bases de datos, como informix, notifican de esta circunstancia y tiran el motor de base de datos a fin de no causar más daño en la integridad transaccional de los datos, previa alarma de que los logical logs se están llenando.

Por si esto no fuera suficiente, el motor quedará atorado en el intento por arrancar al realizar la operación conocida como Fast Recovery. Un proceso de revisión y corrección de posibles inconsistencias transaccionales que comentaremos en un capítulo posterior.

La única manera de levantar el motor en estas condiciones es:

- Recuperando de un respaldo.
- Añadiendo, fuera de línea, más logical logs files (lo que no siempre es posible).



· Cambiando los catálogos internos para que ignore las transacciones pendientes, dejando a la base de datos en un estado probablemente inconsistente.

Como podemos ver esta situación no es en lo absoluto deseable y debe ser evitada a toda costa. De hecho esta situación es PERFECTAMENTE PREVESIBLE Y EVITABLE, las causas que llevan a esto son, generalmente, las siguientes:

· Cuando no se han respaldado los logical logs y por ello no pueden ser liberados: Si uno no está interesado en respaldar los logical logs uno debe configurar el RDBMS para que libere el logical log file en cuanto sus transacciones terminen. O

· Cuando se presenta una long transaction: Entendiéndose por long transaction a aquella transacción que puede llegar a ocupar todos los logical logs; para evitar que una posible long transaction llene los logical logs, casi todos los manejadores de bases de datos tienen niveles, o marcas de agua, de uso del espacio total de los logical log files a nivel de transacción, de llegar una transacción a este nivel (por ejemplo 50%) la transacción es abortada y el mensaje de error de long transaction es devuelto al usuario en tiempo de ejecución. Para evitar que procesos sean abortados por long transaction (LNTX) se debe incrementar el espacio disponible en logical logs files o, si las marcas de agua están muy bajas (30% por ejemplo), se puede incrementar el umbral aunque no es lo más recomendable.



LTXHWM= 50

Por último, aunque decidamos que nuestra base de datos no sea transaccional, aun así existirán operaciones que generen necesariamente transacciones, tal es el caso de la creación de tablas, reestructura de las mismas o borrado. A este tipo de transacciones les llamamos transacciones implícitas.

Transacciones

COMMIT

ROLLBACK

Bitácora de transacciones

Logical Log File

Respaldo de logical logs.

Long Transaction

System Down

Transacciones implícitas.