TEMA 7. INTEGRIDAD, SEGURIDAD Y PRIVACIDAD DE LAS BASES DE DATOS

La integridad de las bases de datos son todos aquellos mecanismos disponibles en el modelo de datos y en el SGBD que permiten que la base de datos se encuentre en un estado consistente con la representación del problema por la que están siendo usadas. Son reglas que se activan en las operaciones de inserción, modificación y borrado de elementos de la base de datos.

Los asertos son predicados que expresan una condición que la base de datos debe satisfacer siempre (create assertion <nombre-aserto> check <predicado>).

Los triggers son acciones que el sistema ejecuta de forma automática antes, en lugar de, o después de que se realiza una operación que pueda ocasionar una modificación de la base de datos. Son mecanismos útiles por el SGBD para alertar a los usuarios o para realizar de manera automática ciertas tareas cuando se cumplen determinadas condiciones. Sus elementos son:

- Evento: Cambio hecho sobre la base de datos que activa el disparador.
- Condición: Solicitud o prueba que se ejecuta cuando se activa el disparador.
- Acción: Procedimiento que es ejecutado cuando el disparador es activado y la condición es verdadera.

```
CREATE TRIGGER <trigger name>
<BEFORE|AFTER> <INSERT|DELETE|UPDATE>
ON <relation name>
FOR EACH <ROW|STATEMENT>
EXECUTE PROCEDURE <procedure name>
(<function args>);
```

El objetivo de la recuperación de la base de datos es proteger la base de datos contra fallos lógicos y físicos que destruyan los datos en todo o en parte. Independiente de la naturaleza de los fallos, éstos pueden afectar a 2 aspectos del almacenamiento de la base de datos: fallos que provocan la pérdida de memoria volátil y fallos que provocan la pérdida del contenido de memoria secundaria.

Una transacción es una secuencia de operaciones que han de ejecutarse en forma atómica (se realizan todas las operaciones que comprende la transacción o no se realiza ninguna). Las transacciones o terminan con éxito y son grabadas en la base de datos, o bien fracasan y debe ser restaurado el estado anterior de la base de datos. Las características de una transacción son:

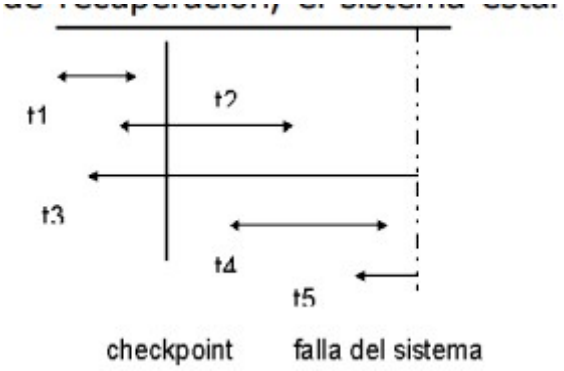
1. Atomicidad (Atomicity): Se ejecutan todas las sentencias o ninguna.
2. Preservación de la consistencia (Consistent): La ejecución de una transacción deja la base de datos en un estado consistente.
3. Aislamiento (Isolation): Una transacción no muestra los cambios que produce hasta que finaliza.

4. Persistencia (Persistent): Una vez finaliza la transacción con éxito, sus efectos perduran en la base de datos.
5. Seriabilidad: El efecto de realizar transacciones concurrentemente debe ser el mismo que se produciría al ejecutarlas por separado en un orden secuencial según van entrando al sistema.

El componente del administrador encargado de lograr la atomicidad se conoce como administrador de transacciones y las operaciones COMMIT (comprometer) y ROLLBACK (retroceder) son la clave de su funcionamiento.

Para conseguir anular y recuperar transacciones, el método más usado consiste en utilizar un archivo de diario o log en el que va guardando toda la información necesaria para deshacer (en caso de fracasar) o rehacer (en caso de recuperar) las transacciones. Este archivo consta de: Identificador de la transacción, Hora de modificación, Identificador del registro afectado, Tipo de acción, Valor anterior del registro, Nuevo valor de registro e Información adicional. Otra alternativa es manejar 2 archivos de log, uno con la imagen anterior a las modificaciones y otro con la imagen posterior a las modificaciones. El archivo log es usualmente una pila que una vez llena va eliminando registros según van entrando nuevos.

El CHECKPOINT permite manejar en forma eficiente el contenido de los archivos log (permiten no tener que recorrer todo el archivo log) ante fallas. El establecimiento de puntos de revisión implica grabar físicamente el contenido de los buffers de datos a las base de datos física, y grabar físicamente un registro de punto de revisión especial dentro del archivo de log o bitácora. Los puntos marcados como CHECKPOINT permiten la recuperación de la base de datos en caliente (después de la caída del sistema se obtiene la dirección del registro de recuperación más reciente y se recorre el archivo de log desde el punto marcado hasta el CHECKPOINT.



La transacción t1 no se ve afectada por la falla del sistema, ni por el proceso de recuperación, por haberse completado antes del último checkpoint. Las transacciones t2 y t4, a pesar de haber terminado no han sido grabadas en la base de datos, ya que serían cometidas en un checkpoint. Las transacciones t3 y t5 deberán rehacerse pues no han concluido. El procedimiento que deberá realizar el sistema al reiniciarse consiste en:

1. Comenzar con 2 listas de transacciones (ANULAR Y REPETIR). Igualar la lista ANULAR a la lista de todas las transacciones incluidas en el registro de checkpoint. Dejar vacía la lista REPETIR.
2. Examinar la bitácora hacia adelante a partir del registro CHECKPOINT.
3. Si se encuentra una entrada de bitácora de “iniciar transacción” para la transacción T, añadir T a la lista ANULAR. Si se encuentra una entrada de bitácora de “comprometer” para la transacción T, pasar esa transacción de la lista ANULAR a la lista REPETIR.
4. Cuando se llegue al final de la bitácora, la lista ANULAR identificará las transacciones t3 y t5, y la lista REPETIR las transacciones t2 y t4.

Posteriormente, el sistema revisará la bitácora hacia atrás, anulando todas las transacciones de la lista ANULAR. A continuación la revisará hacia adelante, realizando de nuevo todas las transacciones en la lista REPETIR. Por último, una vez terminada todas las actividades de recuperación, el sistema estará listo para aceptar nuevos trabajos.

En sistemas multiusuario es necesario un mecanismo para controlar la concurrencia (se pueden producir inconsistencias importantes derivadas del acceso concurrente).

Las técnicas de bloqueo están basadas en una variable asociada a cada elemento de datos que describe el estado de dicho elemento respecto a las posibles operaciones (recuperación o actualización) que se pueden realizar sobre ellos en cada momento. Los tipos de bloqueo pueden ser exclusivos (ninguna otra transacción puede acceder al objeto bloqueado ni bloquearlo hasta que sea liberado por la transacción que lo había retenido. Se utiliza cuando se requiere actualizar datos) o compartidos (permite que otras transacciones retengan también el objeto en bloque compartido, pero no exclusivo. Se utiliza cuando no se quiere actualizar datos pero se desea impedir cualquier modificación mientras los datos son consultados).

El algoritmo utilizado se llama bloqueo de dos fases. El problema de las técnicas de bloqueo es que puede producirse un interbloqueo (deadlock), dos o más transacciones están esperando cada una de ellas que la otra libere algún objeto antes de seguir. Se puede solucionar con algunas técnicas como: prevenir el deadlock (obliga a que las transacciones bloqueen todos los elementos que necesitan por adelantado. En caso de no poder conseguir todos esos elementos, no bloquea ninguno y se queda en espera hasta volver a intentarlo) o detectar el deadlock (se controla de forma periódica si se ha producido un deadlock. Se construye un grafo en espera, cada nodo es una transacción en ejecución y un arco de una transacción Ti a Tj, en caso que ti esté esperando un elemento que ocupa Tj. Si existe un ciclo en el grafo tenemos un deadlock. La solución es escoger transacciones víctimas y deshacerlas, hasta que desaparezca el deadlock. Cada SGBD tiene políticas diferentes para escoger víctimas).

Una granularidad muy gruesa implica gestionar un menor número de bloqueos, pero retrasa la ejecución de muchas transacciones (los objetos no se van liberando). Una

granularidad muy fina permite mayor concurrencia, pero aparecen mas situaciones de deadlock que han de ser resueltas.

En las técnicas de marcas de tiempo, las marcas de tiempo son identificadores únicos que se asignan a las transacciones, que se consideran como el tiempo de inicio de una transacción. Con esta técnica no existen bloqueos, las transacciones se ordenan en función de su marca de tiempo y se ejecutan o se retrasan.

En las técnicas optimistas, las transacciones acceden libremente a los elementos y, antes de finalizar, se determina si ha habido interferencias. Considera que las transacciones tienen 3 fases:

1. Lectura: Las transacciones realizan operaciones sobre copias privadas de los objetos (accesibles solo por la transacción).
2. Validación: Se comprueba si el conjunto de objetos modificados por una transacción se solapa con el conjunto de objetos modificados por alguna otra que haya hecho la validación durante la fase de lectura de dicha transacción.
3. Grabación: En el caso de no detectar interferencias se graban las modificaciones, convirtiendo las versiones privadas en versiones actuales.

La seguridad de las bases de datos es un área amplia que abarca varios temas: cuestiones éticas y legales relativas al derecho de tener acceso a cierta información, cuestiones de política a nivel gubernamental, institucional o corporativo, relacionadas con el tipo de información que no debe estar disponible para el público, y cuestiones relacionadas con el sistema. Las necesidades en las organizaciones de identificar múltiples niveles de seguridad y clasificar los datos y los usuarios según estos niveles.

La seguridad de las bases de datos se refiere a la protección frente a accesos malintencionados. Para proteger la base de datos hay que adoptar medidas de seguridad en varios niveles.

En relación al SGBD, debe mantener información de los usuarios, su tipo y los accesos y operaciones permitidas de éstos. Los SGBD tienen opciones que permiten manejar la seguridad (GRANT, REVOKE, …). También tienen un archivo de auditoría en donde se registran las operaciones que realizan los usuarios. Otro mecanismo de seguridad que ofrecen es entregar información a los usuarios a través de vistas (CREATE VIEW). Los tipos de usuarios son:

- DBA: Están permitidas todas las operaciones, conceder privilegios y establecer usuarios.
- Usuario con derecho a crear, borrar y modificar objetos: Además puede conceder privilegios a otros usuarios sobre los objetos que ha creado.
- Usuario con derecho a consultar o actualizar: Sin derecho a crear o borrar objetos.

Las vistas son un medio de proporcionar a un usuario un modelo personalizado de base de datos. Una vista puede ocultar los datos que un usuario no necesita ver. La capacidad de las vistas para ocultar datos sirve para simplificar el uso del sistema y

para mejorar la seguridad (CREATE VIEW <nombre-vista> AS <condition>). La creación de vistas no necesita la autorización de recursos. El usuario que crea la vista no recibe necesariamente todos los privilegios sobre la misma, sólo recibe los privilegios que no proporcionan autorizaciones adicionales respecto de las que ya posee. Si un usuario crea una vista sobre la que no se puede conceder ninguna autorización, se deniega la solicitud de creación de la vista.

Un ejemplo general de asignación de privilegios sería: grant <lista-privilegios> on <lista-relaciones> to <lista-usuarios>.

Un papel o rol define un tipo de usuario de la base de datos que tiene concedidos una serie de autorizaciones sobre la misma. Los papeles permiten simplificar la concesión de privilegios a los usuarios de forma individualizada, asignándoles un papel que tiene asignados esos privilegios. Un ejemplo general de creación de papeles y asignación de privilegios sería: create role <nombre-papel> grant <lista-privilegios> on <lista-relaciones> to <lista-papeles> grant <nombre-papel> to <lista-usuarios>.

La autenticación se refiere a la tarea de verificar la identidad de una persona o software que se conecta a una base de datos. La forma más simple consiste en una contraseña secreta que se debe presentar cuando se abra una conexión a la base de datos. Otra aplicación son las firmas digitales para verificar la autenticidad de los datos (la clave privada se usa para firmar los datos y los datos firmados se deben hacer públicos. También sirven para asegurar el rechazo).