Ingeniería en Sistemas ula

Bases de datos

Recuperación y seguridad

Técnicas de recuperación
Seguridad y autorización

Recuperación

Descripción conceptual de diversas estrategias para la recuperación ante fallas.

Recuperar = restaurar la BD a un estado anterior correcto cercano al momento de la falla.

Fallas:

1. Caída del sistema (falla del computador)
2. Error en la transacción o en el sistema
3. Errores detectados por la transacción
4. Fallas en el control de concurrencia

Sin daños físicos

5. Falla de disco
6. Falla catastrófica (incendio, inundación)

Recuperación típica

- Si hay daños extensos a la BD por alguna **falla catastrófica**, el método de recuperación restaurará una copia anterior de la BD que estará en almacenamiento secundario y luego se reconstruye un estado mas actualizado *rehaciendo* las operaciones de las transacciones confirmadas asentadas en la bitácora hasta el momento del fallo.

- Si la BD **no tiene daños físicos** pero se ha vuelto inconsistente debido a fallas no catastróficas, el método de recuperación revierte los cambios que provocaron la inconsistencia *deshaciendo* algunas operaciones y quizás sea necesario *rehacer* algunas operaciones para restaurar un estado consistente de la BD.

Recuperación ante fallas no catastróficas

- **Técnicas de actualización diferida**

Actualizan la BD sólo después que una transacción llega a su punto de confirmación, antes de esto todas las actualizaciones se asientan en el espacio local de trabajo local de la transacción. Durante la confirmación de la transacción las actualizaciones se guardan primero en la bitácora y luego se escriben en la BD.

- **Técnicas de actualización inmediata**

Es posible que algunas operaciones de una transacción actualicen la BD antes que la transacción llegue a su punto de confirmación. Sin embargo, estas operaciones se escriben en la bitácora en disco antes de aplicarlas a la BD.

Actualización diferida

- **No deshacer / Rehacer = actualización diferida**

Si una transacción falla antes del punto de confirmación no habrá modificado la BD en nada

NO DESHACER

Quizás sea necesario rehacer el efecto de las operaciones de una transacción confirmada (gracias a la bitácora)

REHACER

Actualización inmediata

- **Deshacer / Rehacer = actualización inmediata**

Si una transacción falla antes del punto de confirmación, es necesario revertir la transacción

DESHACER

Quizás sea necesario rehacer el efecto de las operaciones de una transacción confirmada (gracias a la bitácora)

REHACER

Si todas las actualizaciones se escriben en la BD antes de confirmar la transacción, sólo se necesita deshacer
(**DESHACER/NO REHACER**)

SGBD y caché

Caché del SGBD = Grupo de buffers dentro de memoria principal

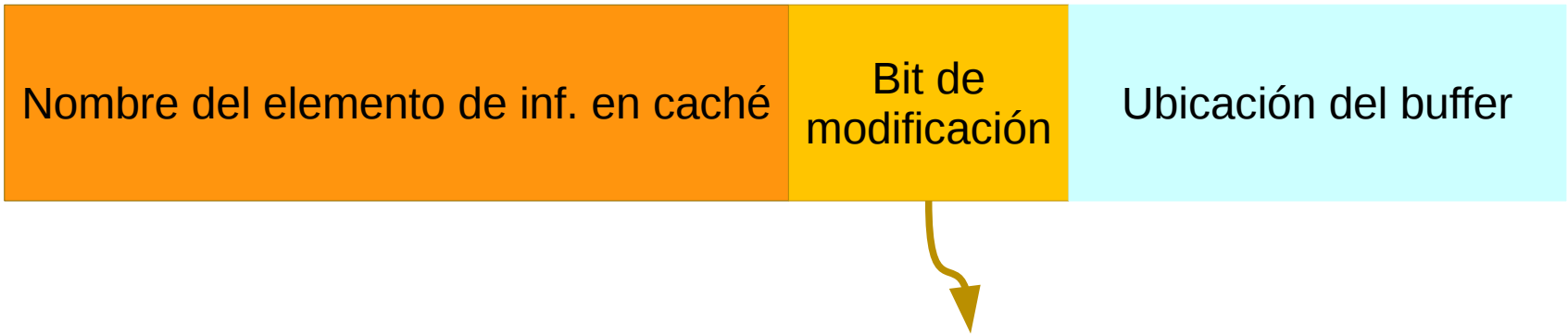
Directorio del caché **<nombre_elemento, ubicación_del_buffer>**

1.- SGBD necesita elemento
2.- SGBD busca elemento en el directorio del caché
3.- Si el elemento no está en el directorio, lo busca en disco y copia sus páginas al caché.

Puede ser necesario **desalojar** algunos buffers de caché para ubicar al nuevo elemento. → LRU (el menos usado)
FIFO (primero en entrar, primero en salir)

SGBD y caché

Directorio del caché



0: no se ha modificado el elemento
1: si se ha modificado el elemento

Cuando se **desaloja** un elemento del caché, se escribirá en disco sólo si su bit de modificación es igual a 1

Desalojo de la memoria caché

- **Actualización en el lugar**: escribe el elemento en la misma ubicación en el disco, sobreescribiendo su valor anterior. Una sola copia de cada elemento en disco.
  Necesita usar una bitácora para la recuperación
  Escritura anticipada en la bitácora en disco

- **Creación de sombras**: escribe el elemento en el disco en un lugar diferente. Se pueden mantener múltiples copias de un mismo elemento.

  Valor antiguo = BFIM (before image)
  Valor nuevo = AFIM (after image)

Estrategias de desalojo de la caché

- **Actualización en el lugar**: escribe el elemento en la misma ubicación en el disco, sobreescribiendo su valor anterior. Una sola copia de cada elemento en disco.
  Necesita usar una bitácora para la recuperación
  Escritura anticipada en la bitácora en disco

- **Creación de sombras**: escribe el elemento en el disco en un lugar diferente. Se pueden mantener múltiples copias de un mismo elemento de información.

  Valor antes de la actualización = BFIM (before image)
  Valor nuevo = AFIM (after image)

Actualización en el lugar

- Necesita usar una bitácora para la recuperación ante fallas.
- El mecanismo de recuperación debe cuidar que la BFIM del elemento de información quede asentado en la bitácora y que esa entrada de la bitácora se guarde en disco antes que se escriba el AFIM en el elemento de información → **Escritura anticipada en la bitácora** (WAL: write-ahead logging)

Tipos de entrada en bitácora

- **Entrada REHACER**: corresponde a una operación de escritura de un elemento de información de una transacción que incluye el valor nuevo (AFIM).

- **Entrada DESHACER**: corresponde a una operación de escritura de un elemento de información de una transacción que incluye el valor antiguo (BFIM). Las operaciones de lectura también se consideran entradas de este tipo.

Listas del subsistema recuperación del SGBD

- **Transacciones activas**: han iniciado pero no se han confirmado aún.
- **Transacciones confirmadas**: todas las transacciones confirmadas desde el último punto de control.
- **Transacciones abortadas**: todas las transacciones abortadas desde el último punto de control.

- **Punto de control**: En la bitácora del SGBD se escribe periódicamente un registro (punto de control) en el punto en que el sistema escribe en la BD (disco) todos los búferes del SGBD que se han modificado.

  El subsistema de recuperación de un SGBD debe decidir en cuáles intervalos toma un punto de control. El intervalo puede medirse en tiempo (cada m minutos) o como un número de transacciones confirmadas desde el último punto de control.

Reversión de transacciones

Gracias a la bitácora podemos conocer las operaciones de la transacción que deben revertirse y los valores antiguos de los elementos de información.

T1	T2	T3
Leer(A)	**Leer(B)**	**Leer(C)**
Leer(D)	**Escribir(B)**	**Escribir(B)**
Escribir(D)	**Leer(D)**	**Leer(A)**
	Escribir(D)	**Escribir(A)**

- T3 no ha terminado antes de la falla por lo que se revierte. Se deshace el paso 2
- T2 se revierte, porque leyó a B después que T3 lo escribió y T3 se revierte. Se deshacen los pasos 6 y 12

Bitácora →

1.[inicio, T3]	6. [escribir,T2,B,2,6]	11. [leer, T2, D]
2. [leer, T3, C]	7. [inicio, T1]	12.[escribir,T2,D,5,6]
3. [escribir,T3,B,8,2]	8. [leer, T1, A]	13. [leer, T3, A]
4. [inicio, T2]	9. [leer, T1, D]	14. Caída del sistema
5. [leer, T2, B]	10.[escribir,T1,D,2,5]	

Técnicas recuperación basadas en AD

Durante la ejecución de las transacciones las escrituras se guardan sólo en la bitácora y en el espacio de trabajo de la transacción. Cuando se confirma y se fuerza la escritura de la bitácora en disco, luego se guardan los cambios en la base de datos.

Protocolo:

- Una transacción no puede modificar la BD antes de llegar a su punto de confirmación.
- Una transacción no se confirma antes de asentar todas sus operaciones en la bitácora y forzar la escritura de la bitácora en disco.

ALGORITMO
NO DESHACER / REHACER

AD en SGBD monousuario

ALGORITMO RAD_1

- Se usa la lista de transacciones confirmadas y la lista de transacciones activas (siempre tiene a lo sumo una transacción por ser monousuario)
- A partir de la bitácora rehacer todas las operaciones de escritura de las transacciones confirmadas *en el orden en que se escribieron* en la bitácora
- Reiniciar las transacciones activas

Rehacer una operación escribir consiste en examinar su entrada en la bitácora y asignar el nuevo valor de *X* al elemento *X* de la BD.

AD en SGBD monousuario

T1	T2
Leer(A)	**Leer(B)**
Leer(D)	**Escribir(B)**
Escribir(D)	**Leer(D)**
	Escribir(D)

Bitácora:
[inicio, T1]
[escribir, T1, D, 20]
[confirmar, T1]
[inicio, T2]
[escribir, T2, B, 10]
[escribir, T2, D, 24]
caída del sistema

Proceso de recuperación:
- [escribir, T1] se rehace
- Las escrituras de T2 se ignoran (no está confirmada antes de la falla)
- Se reinicia T2

AD en SGBD multiusuario

ALGORITMO RAD_M

- Se usa la lista de transacciones confirmadas y la lista de transacciones activas
- A partir de la bitácora rehacer todas las operaciones de escritura de las transacciones confirmadas *en el orden en que se escribieron* en la bitácora
- Reiniciar las transacciones activas

Suposición: control de concurrencia se hace con bloqueo a dos fases y todos los bloqueos de mantienen vigentes hasta que la transacción llegue a su punto de confirmación, después de eso si se liberan los elementos de información.

AD en SGBD multiusuario

T1	T2	T3	T4
Leer(A)	**Leer(B)**	**Leer(A)**	**Leer(B)**
Leer(D)	**Escribir(B)**	**Escribir(A)**	**Escribir(B)**
Escribir(D)	**Leer(D)**	**Leer(C)**	**Leer(A)**
	Escribir(D)	**Escribir(C)**	**Escribir(A)**

Bitácora

1.[inicio, T1]	6. [escribir, T4 B, 16]	11. [inicio, T3]
2. [escribir, T1, D, 20]	7. [escribir, T4, A, 45]	12. [escribir, T3, A, 44]
3. [confirmar, T1]	8. [confirmar, T4]	13. [escribir, T2, D, 8]
4. [punto de control]	9. [inicio, T2]	14. Caída del sistema
5. [inicio, T4]	10. [escribir, T2,B, 12]	

AD en SGBD multiusuario

Bitácora

1.[inicio, T1]	6. [escribir, T4 B, 16]	11. [inicio, T3]
2. [escribir, T1, D, 20]	7. [escribir, T4, A, 45]	12. [escribir, T3, A, 44]
3. [confirmar, T1]	8. [confirmar, T4]	13. [escribir, T2, D, 8]
4. [punto de control]	9. [inicio, T2]	14. Caída del sistema
5. [inicio, T4]	10. [escribir, T2,B, 12]	

Proceso de recuperación:

- T2 y T3 se ignoran (no llegaron a su pto. confirmación)
- T4 se rehace (su pto. de confirmación ocurre después del último pto. de control)

Técnicas recuperación basadas en AI

Protocolo:

- Se obliga a escribir en la BD antes que la transacción esté confirmada, pero antes de escribir en la BD siempre se escribe en la bitácora y se escribe la bitácora en disco.
- Si una T falla antes de llegar a su punto de confirmación, es necesario deshacer su efecto en la BD.

ALGORITMO DESHACER / REHACER

Si la técnica de recuperación asegura que todas las escrituras de una transacción se asientan en la BD antes que se confirme la transacción entonces no hay necesidad de rehacer → DESHACER / NO REHACER

AI en SGBD monousuario

ALGORITMO RAI_1

- Se usa la lista de transacciones confirmadas y la lista de transacciones activas (siempre tiene a lo sumo una transacción por ser monousuario)
- A partir de la bitácora se deshacen todas las operaciones de escritura de las transacciones activas *en orden inverso al que se escribieron* en la bitácora
- A partir de la bitácora rehacer todas las operaciones de escritura de las transacciones confirmadas en *el orden en que se escribieron* en la bitácora

Deshacer una operación escribir consiste en examinar su entrada en la bitácora y asignar el viejo valor de *X* al elemento *X* de la BD.

AI en SGBD multiusuario

ALGORITMO RAI_M

- Se usa la lista de transacciones confirmadas y la lista de transacciones activas.
- A partir de la bitácora deshacer todas las operaciones de escritura de las transacciones activas *en orden inverso al que se escribieron* en la bitácora
- A partir de la bitácora rehacer todas las operaciones de escritura de las transacciones confirmadas en *el orden en que se escribieron* en la bitácora

Deshacer una operación escribir consiste en examinar su entrada en la bitácora y asignar el viejo valor de X al elemento X de la BD.

Recuperación basada en paginación de sombra

Algoritmo de NO DESHACER / NO REHACER

- No requiere uso de una bitácora en SGBD mono-usuario y puede requerir la bitácora en SGBD multiusuario, si el método de control de concurrencia lo amerita.
- Mantiene una tabla de páginas actuales y una tabla de páginas sombra. La primera mantiene las páginas actuales de la BD, la segunda mantiene la versión anterior de las páginas modificadas
- Cuando se efectúa la operación escribir se crea una nueva versión de la página, se anota su dirección en la tabla de páginas actuales y se mantiene la versión anterior sin modificar en la tabla de páginas sombra.

Recuperación basada en paginación de sombra

- Cuando una T se confirma se desechan sus entradas en la tabla de páginas sombra devolviendo dichas páginas a la lista de páginas disponibles
- Para recuperación se desecha la tabla de páginas actuales y se coloca la tabla de páginas sombra como la tabla de páginas actuales

- Desventajas:
  - Manejo de páginas no contiguas en disco
  - Manejo de almacenamiento de páginas en disco si las tablas de páginas son muy grandes y
  - Recolección de basura cuando las T se confirman

Respaldo de BD

- La BD completa y la bitácora del SGBD se copian periódicamente en un medio de almacenamiento económico.
- Los respaldos son útiles ante la recuperación de fallas catastróficas o fallas de disco.

```
$ pg_dump miBD > miBD.sql

$pg_dump [connection-option...] [option...] [dbname]
```

Seguridad en SGBD

El SGBD cuenta con un subsistema de seguridad y autorización que se encarga de garantizar la seguridad de porciones de la BD contra el acceso no autorizado

- Identificar y autorizar a los usuarios para:
  - uso de códigos de acceso y palabras claves
  - exámenes
  - impresiones digitales
  - reconocimiento de voz
  - barrido de la retina, etc.

Autorización: usar derechos de acceso dados por el terminal, por la operación que puede realizar o por la hora del día

Seguridad en SGBD

Control de acceso discrecional:

- Se usa para otorgar y revocar privilegios a los usuarios a nivel de archivos, registros o campos en un modo determinado (consulta o modificación)
- El administrador de la BD asigna el propietario de un esquema, quien puede otorgar o revocar privilegios a otros usuarios en la forma de: consulta o modificación
- A través del uso de la instrucción GRANT OPTION se pueden propagar los privilegios en forma horizontal o vertical

Ejemplo: GRANT SELECT ON Empleado TO codigoUsuario
REVOKE SELECT ON Empleado FROM codigoUsuario

Control de acceso obligatorio:

Sirve para imponer seguridad de varios niveles tanto para los usuarios como para los datos

Control de acceso a nivel de usuario

CREATE USER name [[WITH] option [...]]

donde option puede ser:

```
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

Control de acceso a nivel de relaciones

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO role_specification [, ...] [ WITH GRANT OPTION ]
```

```
GRANT SELECT ON Empleado TO maria;
GRANT INSERT, DELETE ON Empleado, Cargo TO juan;
GRANT INSERT, DELETE ON Empleado, Cargo TO lucia WITH GRANT OPTION;
```

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

```
REVOKE SELECT ON Empleado FROM maria;
```