TEMA 2 – Procesamiento de transacciones

1. Concepto de transacción

Transacción: secuencia de operaciones de acceso a la BD (consulta o actualización) que constituyen una unidad de ejecución.

- ✓ La transacción se procesa como una **operación atómica**.
- ✓ Las **restricciones** de comprueban al final de la transacción.
- ✓ La transacción se **rechaza** si alguna restricción se viola.

Procesar correctamente una transacción significa:

a) Todas las operaciones de la transacción se ejecutan con éxito y sus cambios quedan grabados permanentemente en la BD, **o bien**

b) La transacción no tiene ningún efecto en la BD.

2. Operaciones y estados de una transacción

Operaciones de acceso a datos en una transacción

LEER(X):

1) Determinar la dirección del bloque que contiene X
2) Copiar el bloque del disco a un buffer de MP (si el bloque no está ya en MP)
3) Copiar el elemento de datos X del buffer a la variable X del programa del usuario

ESCRIBIR(X):

1) Determinar la dirección del bloque que contiene o debe contener X
2) Copiar el bloque del disco a un buffer de MP (si el bloque no está ya en MP)
3) Copiar la variable X del programa del usuario al elemento de datos X en el buffer
4) Copiar el bloque actualizado del buffer al disco. No tiene por qué realizarse inmediatamente.

Operaciones adicionales en una transacción

Operaciones de usuario

- inicio: el usuario indica al SGBD el comienzo de la transacción.
- fin (confirmación parcial): el usuario indica al SGBD que la transacción ha finalizado (el usuario da por buena la transacción que ha definido).
- anulación: el usuario anula la transacción que está definiendo.

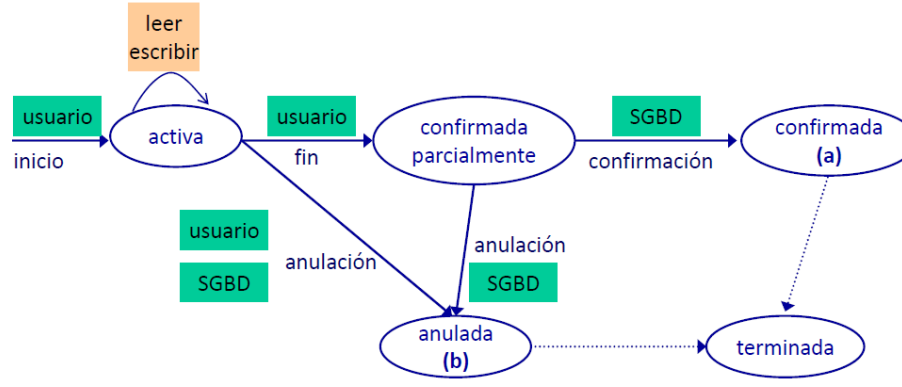
Acciones del SGBD después de la comprobación de las RI

Acciones del SGBD

- confirmación: el SGBD confirma definitivamente la transacción que el usuario ha finalizado con confirmación.
- anulación: el SGBD anula la transacción que el usuario ha finalizado con confirmación.

Estados de una transacción

- **Activa:** el usuario inicia la transacción (**inicio**) y solicita operaciones de lectura o escritura sobre la BD.
- **Confirmada parcialmente:** el usuario finaliza la transacción (**fin**) dándola por buena.
- **Anulada:** el usuario finaliza la transacción anulándola (**anulación**), o bien el SGBD anula la transacción que está activa (por errores) o la que ya ha sido confirmada por el usuario (por la violación de alguna RI).
- **Confirmada:** el SGBD confirma definitivamente una transacción finalizada con confirmación por el usuario tras comprobar que se satisfacen todas las RI.



operaciones del usuario o del SGBD

estados de la transacción

sujeto de la operación

(a) todas las operaciones de la transacción se ejecutan con éxito y sus cambios quedan grabados permanentemente en la BD.

(b) la transacción no tiene ningún efecto en la BD.

3. Propiedades del procesamiento de transacciones

- ✓ **Atomicidad:** una transacción es una unidad de ejecución (o se ejecutan todas sus operaciones o no se ejecuta ninguna de ellas).

- ✓ **Consistencia:** una transacción debe conducir la BD de un estado consistente a otro estado consistente (es decir, que se cumplan todas las RI del esquema).

- ✓ **Aislamiento:** una transacción debe ejecutarse como si se ejecutase de forma aislada.

- ✓ **Persistencia:** los cambios de una transacción confirmada por el SGBD deben quedar grabados permanentemente en la BD.

⬇

Propiedades ACID= Atomicity+Consistency+Isolation+Durability

4. Definición de transacciones en SQL

- ➢ INICIO: **`START TRANSACTION`** (o inicio implícito).
  - El **inicio implícito** se realiza cuando se ejecuta una instrucción SQL (de DML) y no está ninguna transacción activa en ese momento.
- ➢ FIN (confirmación usuario): **`COMMIT [WORK]`**
- ➢ ANULACIÓN (anulación usuario): **`ROLLBACK [WORK]`**
- ➢ ANULACIÓN (con *savepoint*): **`ROLLBACK [WORK] [TO SAVEPOINT marca_savepoint]`**
  - Las marcas de *savepoint* permiten establecer partes opcionales dentro de una transacción que podrán ser posteriormente deshechas sin deshacer la transacción completa.
  - Las marcas de *savepoint* se establecen con **`SAVEPOINT marca_savepoint`**

```
START TRANSACTION
...
...
SAVEPOINT marca1
        ...
        ...
        SAVEPOINT marca2
                ...
                ...
                IF ... THEN ROLLBACK TO SAVEPOINT marca2
        ...
        ...
        IF ... THEN ROLLBACK TO SAVEPOINT marca1
...
COMMIT
```

- **`SET TRANSACTION modo [,modo]`**

  `modo := nivel de aislamiento | modo de acceso | área de diagnóstico`

  Sirve para dar directrices al SGBD sobre el procesamiento de transacciones durante una sesión de usuario.

  Esta instrucción se debe ejecutar entre transacciones y el **alcance** de la directriz es la transacción siguiente.

  Los posibles argumentos son:

  - **Nivel de aislamiento:** nivel de control de la concurrencia que debe realizar el SGBD.

  - **Área de diagnóstico:** número máximo de condiciones de diagnóstico que pueden registrarse relativas a la ejecución de las últimas instrucciones SQL:

    `SET TRANSACTION DIAGNOSTICS SIZE` ***`número`***

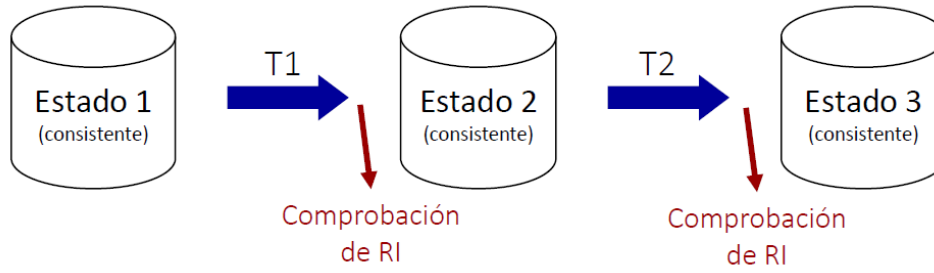
  - **Modo de acceso:** tipo de operaciones que se pueden ejecutar en una transacción: **`READ ONLY`** prohíbe operaciones de actualización de la BD.

    `SET TRANSACTION {READ ONLY | READ WRITE}`

5. Concepto de restricción de integridad

Restricción de integridad (RI): propiedad que la BD debe satisfacer en cualquier instante de su historia.

- ✓ La BD evoluciona por la ejecución de transacciones de usuario.
- ✓ Las transacciones se consideran unidades de ejecución (**atomicidad**).
- ✓ Las restricciones de integridad se deben comprobar después de la ejecución de cada transacción (**consistencia**).



La comprobación de restricciones de integridad tiene que ver con las propiedades de **Atomicidad** y **Consistencia** del principio ACID de la ejecución de transacciones.

6. Comprobación de restricciones en SQL

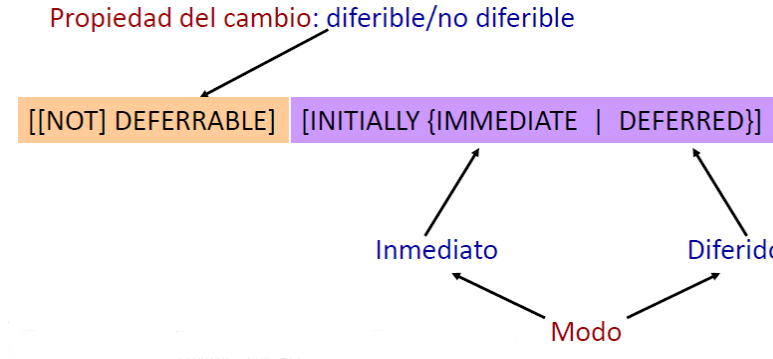
En teoría las RI se deben comprobar cuando termina una transacción, pero en la práctica esta comprobación se puede relajar.

Para ello, cada RI del esquema tiene dos propiedades:

- ✓ **Modo:** define cuando se comprueba la restricción y qué se hace si se viola.
  - **Inmediato:** se comprueba después de cada instrucción SQL que pueda violar la restricción. Si se viola, el SGBD anula la dicha instrucción y la transacción continua.
  - **Diferido:** se comprueba después de cada transacción que contenga una operación SQL que pueda violar la restricción. Si se viola, el SGBD anula toda la transacción.
- ✓ **Propiedad de cambio:** determina la posibilidad de cambiar o no el modo diferido.
  - **Diferible:** el modo de una restricción se puede cambiar durante la ejecución de una transacción. El cambio es local a la transacción, no modifica la definición de la RI.
  - **No diferible:** el modo de una restricción no se puede cambiar durante la ejecución de una transacción.

Sintaxis en SQL

La cláusula ***cuándo_comprobar*** en SQL nos permite establecer el modo y la propiedad de cambio de una RI al momento de definirla.



Si no se especifica nada en la definición, la restricción se define como no diferible y con modo inmediato (valor por defecto).

- ➢ *Si sólo se establece una de las dos partes, la otra toma el valor por defecto correspondiente*.

NOTA: la versión **NOT DEFERRABLE INITIALLY DEFERRED** está prohibida, ya que es contradictoria.

Cambio de modo

La instrucción SQL que permite cambiar, **localmente en una transacción**, el modo de una restricción definida como diferible es:

```
SET CONSTRAINT {restricción1, ... | ALL} {IMMEDIATE | DEFERRED}
```

- ✓ Cada restricción especificada en la lista debe ser **diferible** y la opción **ALL** hace referencia a todas las restricciones diferibles del esquema de la BD.

- ✓ El **alcance** del cambio producido por la instrucción **SET CONSTRAINT** es la transacción en la que se incluye o el fragmento de transacción hasta la siguiente aparición de la instrucción.

- ✓ Si se incluye la instrucción en medio de la transacción con la opción **IMMEDIATE**, las restricciones afectadas por la instrucción son comprobadas cuando se ejecuta ésta, si alguna de estas restricciones falla, la instrucción **SET** falla y el modo de las restricciones permanece sin modificar.