

# Recuperación

Bibliografía: Introducción a los Sistemas de Bases de Datos  
Date, C.J.

## Recuperación de transacciones

- Está vinculado a la noción de **procesamiento de transacciones**.
- Operaciones de **SQL**
  - **COMMIT** (comprometer)
  - **ROLLBACK** (retroceder).
- **Transacción** es una **unidad lógica de trabajo**.

### Ejemplo:

Supongamos que:

- la **tabla P** contiene un campo más: **CANTTOTAL** (cant. total enviada de cada parte).
- Es decir, **CANTTOTAL** para una parte dada es la **suma** de todos los valores **CANT** de todos los registros de la **tabla SP** correspondientes a esa parte.

Consideremos la secuencia de operaciones para añadir un **nuevo envío** a la BD:  
**(S5,P1,1000)**

- **INSERT** agrega el **nuevo envío** a la tabla **SP**
- **UPDATE** actualiza en forma apropiada el campo **CANTTOTAL** de la **parte P1**

Consideremos la secuencia de operaciones para añadir un **nuevo envío** a la BD:  
**(S5,P1,1000)**

```
EXEC SQL WHENEVER SQLERROR GO TO ANULAR;  
EXEC SQL INSERT  
    INTO SP (S#, P#, CANT)  
    VALUES ('S5', 'P1', 1000);  
EXEC SQL UPDATE P  
    SET CANTTOTAL = CANTTOTAL + 1000  
    WHERE P# = 'P1';  
EXEC SQL COMMIT;  
    GO TO TERMINAR;  
ANULAR:  
EXEC SQL ROLLBACK;  
TERMINAR: RETURN;
```

- Una operación supuestamente individual → dos modificaciones de la BD.

**Definición:**

Una **unidad lógica de trabajo** no es una sola operación, sino una **secuencia de varias operaciones** con la cual un **estado consistente** de la BD **se transforma** en otro **estado consistente**, sin conservar por fuerza la consistencia en todos los puntos intermedios.

- En el ejemplo **no se puede permitir una ejecución de una** de las modificaciones **y no de la otra**

→ la BD quedaría **inconsistente**

- Existe la **posibilidad de falla**:
  - por ejemplo: el sistema podría caer luego de ejecutar la primera modificación.

- Si el sistema maneja el **procesamiento de transacciones**, garantizará que:

– si la transacción ejecuta alguna de las modificaciones y después presenta una falla, antes de que llegue a término normal de la transacción, **se anularán esas modificaciones**.

- La **transacción se lleva a cabo en su totalidad, o se cancela en su totalidad** (como sí jamás se hubiera ejecutado)

## COMMIT

- El **componente del sistema** encargado de lograr esa atomicidad se conoce como ***manejador de transacciones***

- y las operaciones

- **COMMIT** (comprometer)
- **ROLLBACK** (retroceder)

de SQL son la clave de su funcionamiento.

- señala el término **exitoso** de la transacción.
- Le dice al manejador de transacciones que **ha finalizado con éxito una unidad lógica**,
- que la **BD** está de nuevo en un **estado consistente** y
- que **se pueden comprometer o hacer permanentes** todas las **modificaciones** efectuadas.

## ROLLBACK

- señala el término **no exitoso** de la transacción.
- Le dice al manejador de transacciones que algo salió mal,
- que la **BD** podría estar en un **estado inconsistente**
- y que todas las **modificaciones** deben **retroceder o anularse**.

## Cómo es posible anular una modificación

- El **sistema mantiene** una **bitácora** o diario donde se registran los **detalles** de todas las **operaciones de actualización**, en particular los **valores inicial y final** del objeto modificado.
- Si resulta necesario **anular** alguna modificación, el sistema puede utilizar la entrada correspondiente de la bitácora para restaurar el valor original del objeto modificado.

Definición:

- Un **punto de sincronización** representa un **límite** entre **dos transacciones** consecutivas,
  - de modo que corresponde al **final** de una unidad de trabajo,
  - y por tanto al punto en el cual la **BD** está (o debería estar) en un **estado de consistencia**.

- Las únicas **operaciones** que establecen un **punto de sincronización** son

- COMMIT
- ROLLBACK
- el inicio de un programa

Cuando se establece un punto de sincronización:

- Se **comprometen** (COMMIT) o se **anulan** (ROLLBACK) todas las modificaciones realizadas por el programa desde el punto de sincronización anterior
- Se **cierran** los **cursores** abiertos y se pierde todo posicionamiento en la BD
- Se **liberan** todos los **registros bloqueados**

- Una sola ejecución de programa puede incluir varias transacciones.
- Cada COMMIT o ROLLBACK termina una transacción o inicia la siguiente.
  - ➔ las transacciones no son sólo la *unidad de trabajo* sino también la *unidad de recuperación*.
- Si una transacción se compromete con éxito, el sistema deberá garantizar el establecimiento permanente de sus modificaciones en la BD, aún si el sistema cae en el instante siguiente.

### Regla: *protocolo de bitácora de escritura adelantada*

- La bitácora se deberá grabar físicamente antes de poderse completar el procesamiento de una instrucción COMMIT.
- Así el procedimiento de reinicio recuperará todas las transacciones completadas con éxito pero cuyas modificaciones no lograron grabarse físicamente antes de la caída.
  - ➔ las transacciones son la *unidad de recuperación* y también son la *unidad de concurrencia*

### Recuperación del sistema y de los medios de almacenamiento

- El **sistema** debe estar **preparado para fallas globales**,
  - por ejemplo corte de energía.
- Una falla **local** afecta solo la **transacción**
- Una falla **global** afecta a **varias o todas**

### Recuperación del sistema y de los medios de almacenamiento

#### Categorías de fallas

- **Fallas del *sistema*:**
  - afectan a **todas** las **transacciones** que se están realizando **pero no dañan físicamente a la BD**.
    - Ejemplo: interrupción de electricidad.
- **Fallas de los *medios de almacenamiento*:**
  - **dañan físicamente a la BD**.
    - Ejemplo: rotura del disco.

### Fallas del sistema

- **Punto crítico:** se **pierde** el **contenido** de la **memoria principal** (áreas de almacenamiento temporal: buffers de datos).
- Por lo tanto: ya **no se conoce** el **estado** preciso de la **transacción** que se realizaba en el **momento de la falla**.

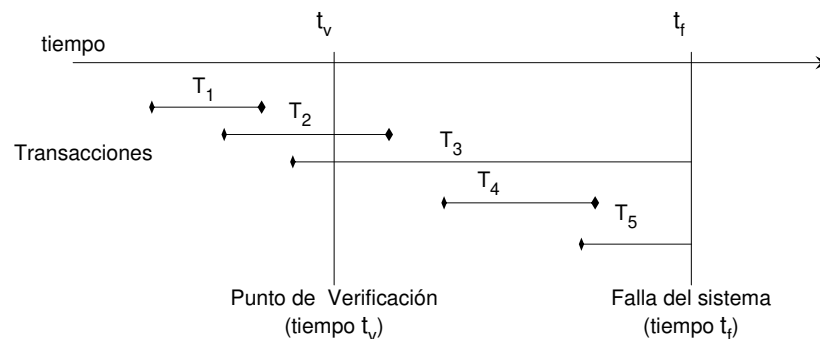
## Fallas del sistema

- Esa **transacción no se podrá completar con éxito**.
  - Es preciso **anularla** (retroceder cuando se reinicie el sistema)
- Podría ser necesario **volver a realizar transacciones al reiniciar el sistema**
  - (aquellas cuyas modificaciones no lograron ser transferidas de los buffers de la BD a la BD física)

*¿cómo sabe el sistema cuáles anular y cuáles realizar otra vez?*

- Cada cierto intervalo previamente **establecido**, el sistema “**establece un punto de revisión**” de manera automática.
- Esto implica **grabar físicamente**:
  - el **contenido de los buffers** en la **BD física**
  - un **registro de punto de revisión especial** en la **bitácora física**
- El **registro de punto de revisión incluye** una **lista** de todas las **transacciones** que se estaban realizando en el momento de establecerse el punto de revisión.

## Ejemplo de uso de esta información:



- Se presentó una **falla de sistema** en el momento **tf**
- El **punto de verificación** más reciente antes de  $t_f$  se tomó en el momento **tv**
  - Las transacciones del tipo  $T_1$  se **completaron** antes del  $t_v$
  - Las transacciones del tipo  $T_2$  se **iniciaron** antes del tiempo  $t_v$  y se **completaron** después del tiempo  $t_v$  y antes del  $t_f$
  - Las transacciones del tipo  $T_3$  se **iniciaron** antes del tiempo  $t_v$  y se pero **no se completaron** antes del  $t_f$
  - Las transacciones del tipo  $T_4$  se **iniciaron después** del tiempo  $t_v$  y se **completaron** antes del  $t_f$
  - Las transacciones del tipo  $T_5$  se **iniciaron después** del tiempo  $t_v$  pero no se **completaron** antes del  $t_f$

Al **reiniciarse** el sistema:

- Deberán **anularse** las transacciones del tipo **T3 y T5**
- Deberán **realizarse de nuevo** las transacciones del tipo **T2 y T4**
- Las del tipo **T1** no entran porque sus **modificaciones ya fueron grabadas físicamente** en la BD en el momento tv **como parte del proceso de punto de revisión**.

## Fallas de los medios de almacenamiento

- Se **destruye físicamente** la BD

Esto implica:

- **cargar de nuevo o restaurar** la BD a partir de un **respaldo** y
- **utilizar** después la **bitácora** (activa y archivada) para realizar de nuevo todas las transacciones desde que se hizo la copia de respaldo.