

# *Administración de Transacciones*



*Andrés Moreno S.*

---

---

---

---

---

---

---

---

## *Protección de los Datos*

- ❏ Recuperación y Concurrencia son los dos problemas más importantes que abordan las Bases de Datos, para la protección de los datos.

2

---

---

---

---

---

---

---

---

## *Recuperación*

- ❏ Implica la recuperación de la propia base de datos, restableciendo la base de datos a un estado correcto (consistente)
- ❏ Esto se debe mayormente a la redundancia

3

---

---

---

---

---

---

---

---

## ***Recuperación***

- ❏ Garantizado la reconstrucción de la información de la BD, a través de la información reconstruida a partir de información guardada redundantemente en algún lugar distinto del sistema.
- ❏ Sistemas grandes de Administración de Bases de Datos (DBMS), {Información compartida y multiusuarios}

4

---

---

---

---

---

---

---

---

## ***Transacciones***

- ❏ Unidad de trabajo lógica que es completada íntegramente o no se realizada ninguna parte de ella.
- ❏ La ejecución de un programa que consulta y/o modifica el contenido de la BD recibe el nombre de transacción.
- ❏ Si las operaciones no modifican ningún dato, sino que sólo recuperan datos, la transacción es de sólo-lectura.

5

---

---

---

---

---

---

---

---

## ***Transacciones***

- ❏ Las transacciones más conflictivas pero más habituales que estudiaremos en detalle, son las que modifican la BD (escritura).
- ❏ Las transacciones no pueden violar ninguna restricción de integridad de la BD, es decir, si la BD era consistente cuando empezó una transacción, también debe serlo cuando termine con éxito la transacción.

6

---

---

---

---

---

---

---

---

## *Operaciones de las Transacciones*

- 🕒 Lectura
  - 🕒 Leer\_ítem(X): Lee el ítem X y almacena su valor en una variable.
- 🕒 Escritura
  - 🕒 Escribir\_ítem(X): Graba el valor de la variable de programa X en el ítem X.

7

---

---

---

---

---

---

---

---

## *Ítems de datos*

- 🕒 La BD se considera subdividida en pequeñas partes (ítems ó gránulos) que pueden reservarse o bloquearse (lock) por una transacción, y posteriormente liberarse.

8

---

---

---

---

---

---

---

---

## *Ítems de datos*

- 🕒 Cuando una transacción reserva un ítem, puede impedir que otras transacciones accedan al ítem hasta que quede liberado. Una parte del DBMS, llamada Gestor de Bloqueos (Lock Manager), asigna y reserva los ítems, así como resuelve dos o más peticiones de reserva sobre el mismo ítem.

9

---

---

---

---

---

---

---

---

## *Tamaño de los Ítems*

- ❏ La naturaleza y el tamaño de los ítems los debe elegir el diseñador del sistema.
- ❏ En el modelo relacional se pueden escoger ítems grandes o pequeños, disponibles por medio del bloqueo de tablas o de filas respectivamente.

10

---

---

---

---

---

---

---

---

## *Tamaño de los Ítems*

- ❏ La elección de ítems grandes disminuye la sobrecarga del sistema debido al mantenimiento, ya que se tienen que tomar menos acciones con respecto a los bloqueos.
- ❏ Sin embargo, la elección de ítems pequeños permite que muchas transacciones operen en paralelo, ya que hay menos probabilidad de que reserven los mismos ítems.

11

---

---

---

---

---

---

---

---

## *Necesidad Control de Concurrency*

- ❏ Al ejecutar transacciones concurrentes se pueden producir estados inconsistentes en la BD.
- ❏ Como ejemplo, comentamos el caso de la BD para gestionar las reservas de una compañía aérea.

12

---

---

---

---

---

---

---

---

## Ejemplo de Concurrency

- ❏ Para cada vuelo se incluye un registro que, entre otros datos almacena el número de asientos reservados.
- ❏ Transacción 1 (T1): cancela N reservas desde un vuelo que tiene X asientos reservados y las cambia a otro vuelo que tiene Y asientos reservados
- ❏ Transacción 2 (T2): reserva M asientos adicionales a los X ya reservados en el primer vuelo.

---

---

---

---

---

---

---

---

## Ejemplo de Concurrency

- |                   |                   |
|-------------------|-------------------|
| ❏ T 1             | ❏ T 2             |
| Leer_item(X);     | Leer_item(X);     |
| X := X-N;         | X := X+M;         |
| Escribir_item(X); | Escribir_item(X); |
| Leer_item(Y);     |                   |
| Y := Y+N;         |                   |
| Escribir_item(Y); |                   |

14

---

---

---

---

---

---

---

---

## Tipos de Fallas

- ❏ El computador falla (falla del sistema)
- ❏ Falla del disco
- ❏ Falla de memoria
- ❏ Un error de sistema en la transacción
- ❏ Errores locales o condiciones de excepción detectados por la transacción
- ❏ Imposiciones del control de concurrencia
- ❏ Problemas físicos y catástrofes (fallas de corriente)

15

---

---

---

---

---

---

---

---

## *Operaciones de Transacción*

- ❏ Para propósitos de recuperación, el sistema necesita conocer cuando una transacción comienza, aborta o termina. El Gestor de Recuperaciones debe conocer la situación de las siguientes operaciones: BEGIN, READ, END, COMMIT y ROLLBACK.

16

---

---

---

---

---

---

---

---

## *Operaciones de Transacción*

- ❏ BEGIN: Esta operación marca el comienzo de la ejecución de la transacción.
- ❏ READ ó WRITE: Especifican operaciones de lectura y/o escritura de ítems que son ejecutadas como parte de una transacción.
- ❏ END: Indica que las operaciones de lectura y escritura de la transacción han concluido y marca el fin de la ejecución de la transacción. **(Es necesario chequear los cambios realizados)**

17

---

---

---

---

---

---

---

---

## *Operaciones de Transacción*

- ❏ COMMIT: Señala un final adecuado de la transacción y que cualquier modificación de ítems ejecutada por la transacción puede ser transferida definitivamente a la BD y no será revertida.
- ❏ ROLLBACK (ABORT): Indica que la transacción ha concluido de forma inadecuada y por tanto, cualquier cambio o efecto que la transacción haya podido tener sobre la BD debe ser revertido.

18

---

---

---

---

---

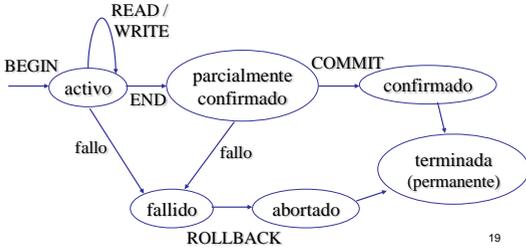
---

---

---

## Diagrama de Transición de Estados

### COMMIT y ROLLBACK puntos de confirmación



19

---

---

---

---

---

---

---

---

## Recuperación de las Transacciones

- Los sistemas de BD mantienen una bitácora, diario (navegación), o log.
- Se guardan todas las instrucciones de las actualizaciones, transacciones y sus procesos.
- El sistema puede utilizar estos registros para restaurar la información de la BD a un punto consistente.

20

---

---

---

---

---

---

---

---

## Importancia de la Información



"I am dressed for success! Of course, my idea of success may not be exactly the same as yours."

21

---

---

---

---

---

---

---

---

## Propiedades de las Transacciones

- ❏ Hay varias propiedades que las transacciones deben poseer por necesidades de los métodos de control de concurrencia y recuperación del DBMS, aunque algunos métodos no precisan de todas las propiedades.
- ❏ Las propiedades deseables de las transacciones son: Atomicidad, Preservación de la Consistencia, Durabilidad o permanencia, Aislamiento, Serializabilidad.

22

---

---

---

---

---

---

---

---

## Propiedades de las Transacciones

- ❏ **Atomicidad:** Una transacción es una unidad atómica de procesamiento; es realizada íntegramente, o no es realizada en absoluto.
- ❏ **Preservación de la Consistencia:** Una correcta ejecución de una transacción debe pasar la BD desde un estado consistente a otro también consistente.
- ❏ **Durabilidad o Permanencia:** Cuando una transacción cambia, la BD y los cambios son confirmados, estos cambios nunca deben perderse por fallos posteriores.

23

---

---

---

---

---

---

---

---

## Propiedades de las Transacciones

- ❏ **Aislamiento:** Una transacción no deberá hacer visibles sus modificaciones a otras transacciones hasta que esté confirmada. Esta propiedad es aplicada más o menos estrictamente por diferentes métodos de control de concurrencia y recuperación.
- ❏ **Serializabilidad:** Este criterio es obligatorio para muchos métodos de control de concurrencia. Varias transacciones son serializables si el efecto de ejecutarlas con operaciones intercaladas es equivalente a ejecutarlas serialmente.

24

---

---

---

---

---

---

---

---

*Administración de  
Transacciones*



*Andrés Moreno S.*

---

---

---

---

---

---

---