

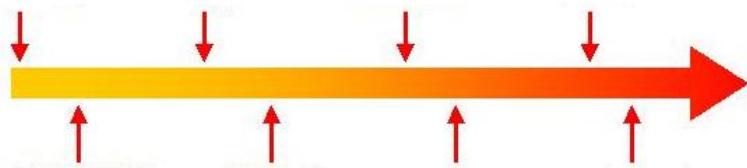


# Sistemas de Bases de Datos 2

2024

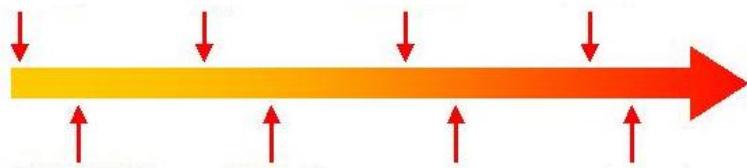
**Ing. Luis Alberto Arias Solórzano**

Unidad 1



# Transacción

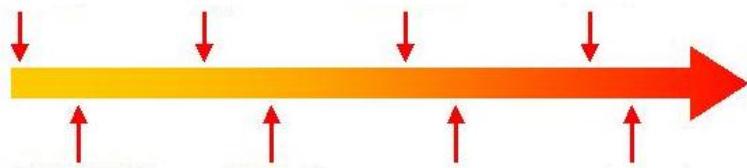
- Una transacción es una *unidad de trabajo lógica* que comprende por lo regular varias operaciones de base de datos.
- El usuario debe ser capaz de informar al sistema cuando haya operaciones distintas que forman parte de la misma transacción. Para este fin, se proporcionan las operaciones BEGIN TRANSACTION, COMMIT y ROLLBACK (iniciar transacción, confirmar y deshacer).
- En esencia, una transacción comienza cuando se ejecuta una operación BEGIN TRANSACTION y termina cuando se ejecuta una operación COMMIT o ROLLBACK correspondiente.



# Transacción

## PROPIEDADES ACID

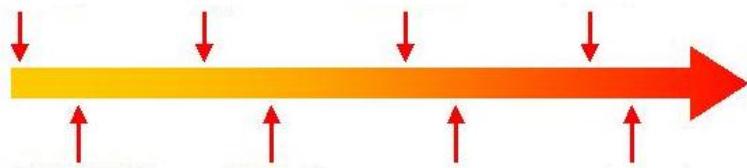
- **Atomicidad.** O todas las operaciones de la transacción se realizan adecuadamente en la base de datos o ninguna de ellas.
- **Consistencia.** La ejecución aislada de la transacción conserva la consistencia de la base de datos.
- **Aislamiento.** Aunque se ejecuten varias transacciones concurrentemente, cada transacción ignora al resto de las transacciones que se ejecuten concurrentemente en el sistema.
- **Durabilidad.** Tras la finalización con éxito de una transacción, los cambios realizados en la base de datos permanecen, incluso si hay fallos en el sistema.



# Transacción

## CARACTERISTICAS

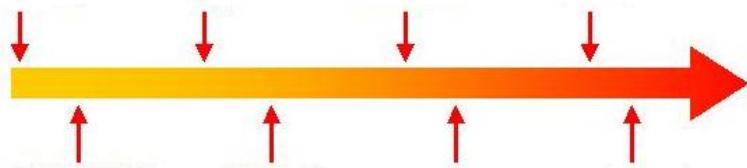
- 1. Se garantiza que las transacciones sean **atómicas**; es decir, se garantiza (desde un punto de vista lógico) que se ejecuten en su totalidad o que no se ejecuten en lo absoluto, aun cuando (por decir algo) el sistema fallara a la mitad del proceso.
- 2. También se garantiza que las transacciones sean **durables**, en el sentido de que una vez que una transacción ejecuta con éxito un COMMIT, debe garantizar que sus actualizaciones sean aplicadas a la base de datos, aun cuando el sistema falle en cualquier punto.



# Transacción

## CARACTERISTICAS

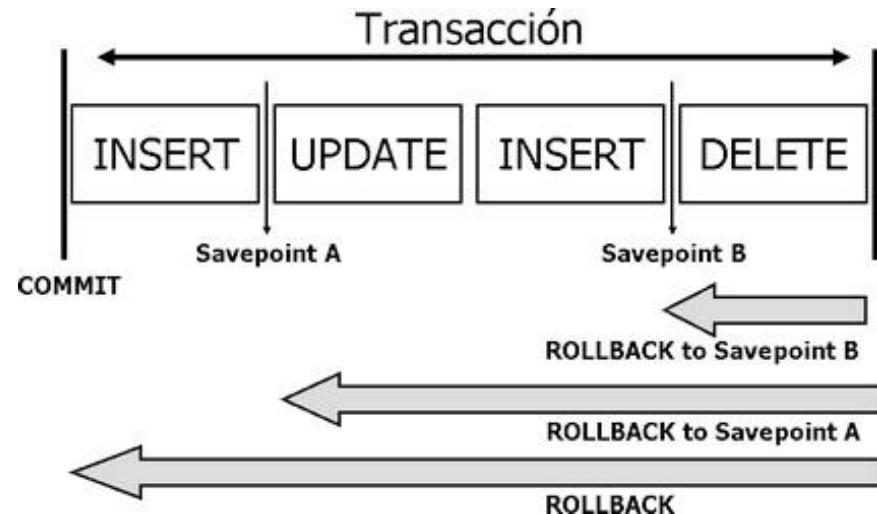
- 3. Se garantiza además que las transacciones estén **aisladas** entre sí, en el sentido de que las actualizaciones hechas a la base de datos por una determinada transacción  $T_1$  no sean visibles para ninguna transacción distinta  $T_2$ , por lo menos hasta que  $T_1$  ejecute con éxito el COMMIT. La operación COMMIT hace que las actualizaciones efectuadas por una transacción sean visibles para otras transacciones; se dice que dichas actualizaciones están *confirmadas* y se garantiza que nunca sean canceladas. Si en vez de ello la transacción ejecuta un ROLLBACK, se cancelan (*deshacen*) todas las actualizaciones hechas por la transacción. En este último caso el efecto es como si nunca se hubiese realizado la transacción.

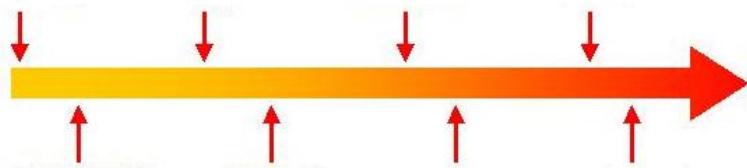


# Transacción

## CARACTERISTICAS

- 4. Se garantiza (por lo regular) que la ejecución intercalada de un conjunto de transacciones concurrentes sea **serializable**, en el sentido que produzca el mismo resultado que se obtendría si se ejecutaran esas mismas transacciones, una a la vez, en un cierto orden serial no especificado.



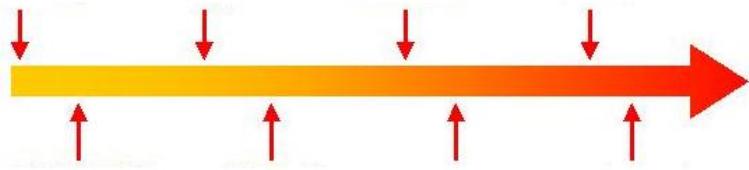


# Transacción

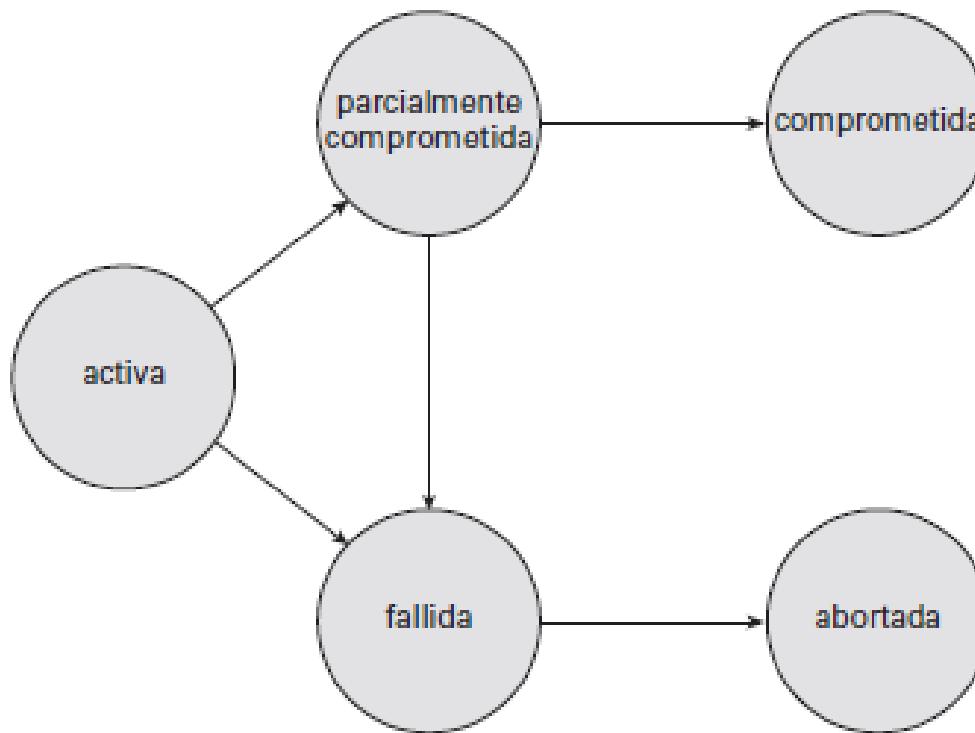
## ESTADOS

- **Activa**, el estado inicial; la transacción permanece en este estado durante su ejecución.
- **Parcialmente comprometida**, después de ejecutarse la última instrucción.
- **Fallida**, tras descubrir que no puede continuar la ejecución normal.
- **Abortada**, después de haber retrocedido la transacción y restablecido la base de datos a su estado anterior al comienzo de la transacción.
- **Comprometida**, tras completarse con éxito.

ESTADOS



# Transacción



# Bitácora de transacciones

- Se utiliza para registrar un historial de cada modificación que sufre la base de datos como resultado de las transacciones.
- Dicho registro es de vital importancia para mantener la *integridad* de los datos y poder deshacer los cambios resultantes de transacciones incompletas ya sea por error del sistema o por la cancelación por parte de los usuarios.
- Durante la operación de la base de datos la escritura a la bitácora tiene prioridad, es decir, todos los cambios primero se escriben a la bitácora y luego se aplican a la base de datos.
- En muchos dbms incluye todo tipo de consulta incluyendo aquellas que no modifican los datos.

# TAREA

Tarea: Investigar la bitácora de transacciones de SQL Server y Oracle, como se puede ver y que guarda.

# Puntos de Sincronización

- Mantener actualizada la información en todo momento, cada vez que cambia un dato, supone una carga computacional excesiva. Una solución consiste en establecer *puntos de sincronización* en los que realizar la actualización. De este modo, si ha sido el último punto de sincronización alcanzado, se debe garantizar que la información contenido base es consistente con la información almacenada.
- La ejecución de una operación COMMIT o ROLLBACK establece lo que se conoce como un *punto de sincronización*. Un punto de sincronización representa el límite entre 2 transacciones consecutivas, de modo que corresponde al final de una unidad lógica de trabajo, y por tanto al punto en el cuál la BD está en un estado de consistencia.
- 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 todos los cursos abiertos y se pierde todo lo posicionamiento en la BD;
  - Se liberan todos los registros bloqueados.
- **Elección de los puntos de sincronización**

La opción más viable con respecto a los puntos de sincronización consiste en utilizar para tal fin la terminación de cada operación de edición o transacciones.



Gracias