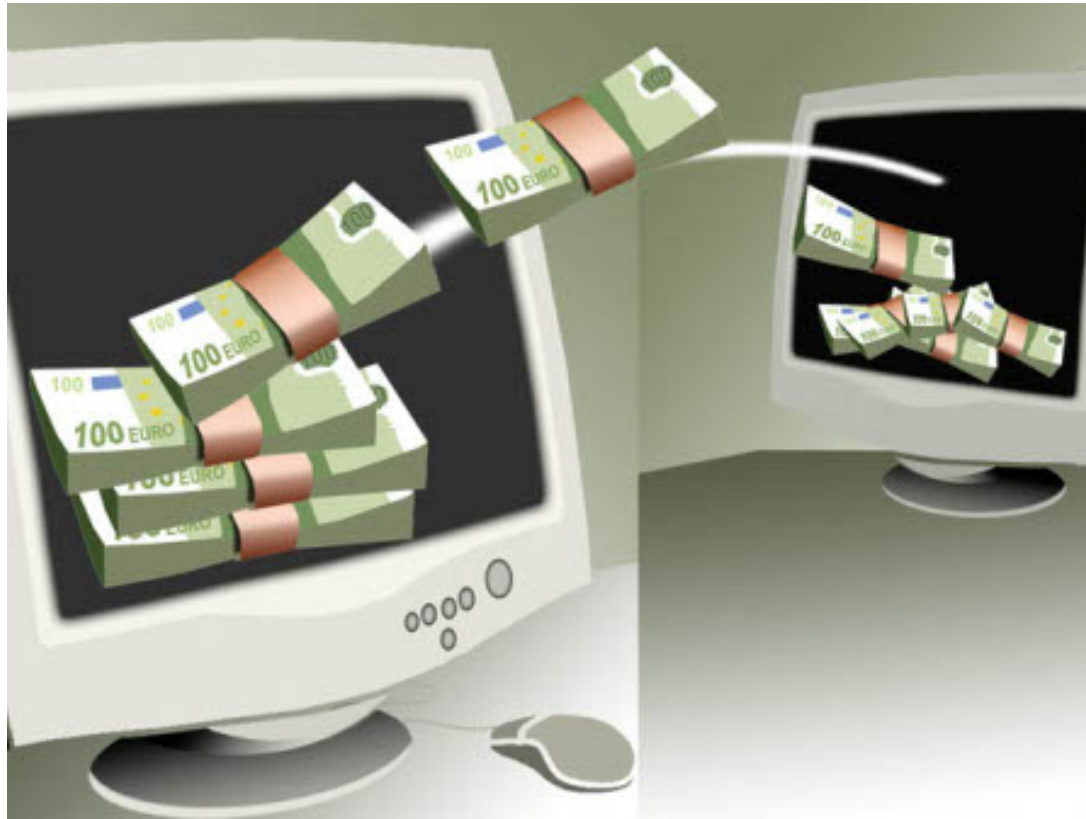


# TRANSACCIONES EN BASES DE DATOS DISTRIBUIDAS

---

Prof. Alejandro Reyes Ortiz

Base de datos distribuidas,  
Universidad Autónoma Metropolitana



# Objetivos

- Explicar el concepto de transacción en base de datos distribuidas.
- Identificar las condiciones de terminación de una transacción.
- Identificar los tipos de transacciones distribuidas y la bitácora.
- Identificar y explicar las cuatro propiedades de las transacciones.

# Transacción

“Una transacción es una secuencia de una o más operaciones agrupadas como una unidad”

El inicio y el final de la transacción definen los puntos de consistencia de la base de datos distribuida. *Si una acción de la transacción no se puede ejecutar, entonces ninguna acción dentro de la secuencia que conforma la transacción tendrá efecto.*

Las operaciones dentro de una transacción se almacenan temporalmente, no a nivel de disco. Es hasta que termina la transacción que se tienen efecto de manera permanente o no.

# Transacciones

- Proviene de los sistemas de Gestión de BD. En el contexto de las bases de datos distribuidas:

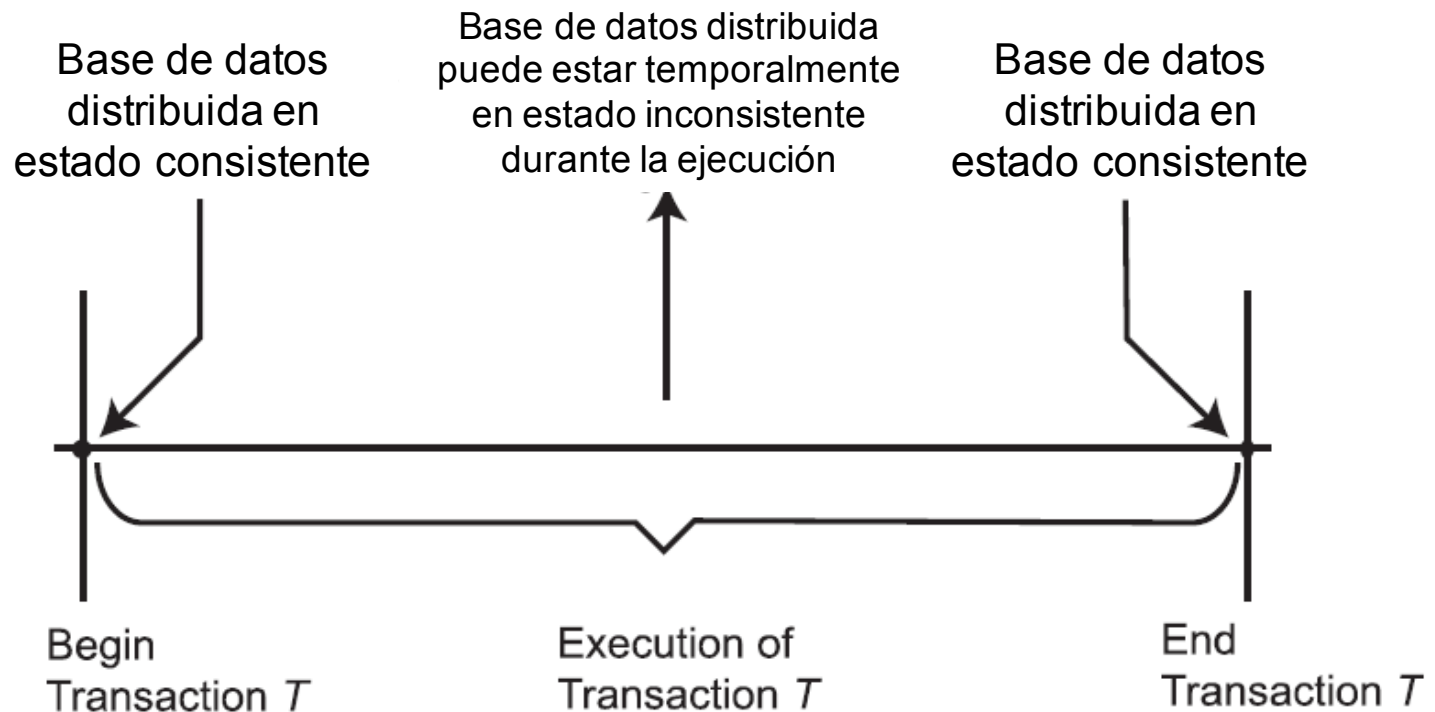
*“Una transacción es la ejecución consistente y confiable de un conjunto de operaciones agrupadas como una unidad que acceden a una base de datos compartida”*

# Transacciones

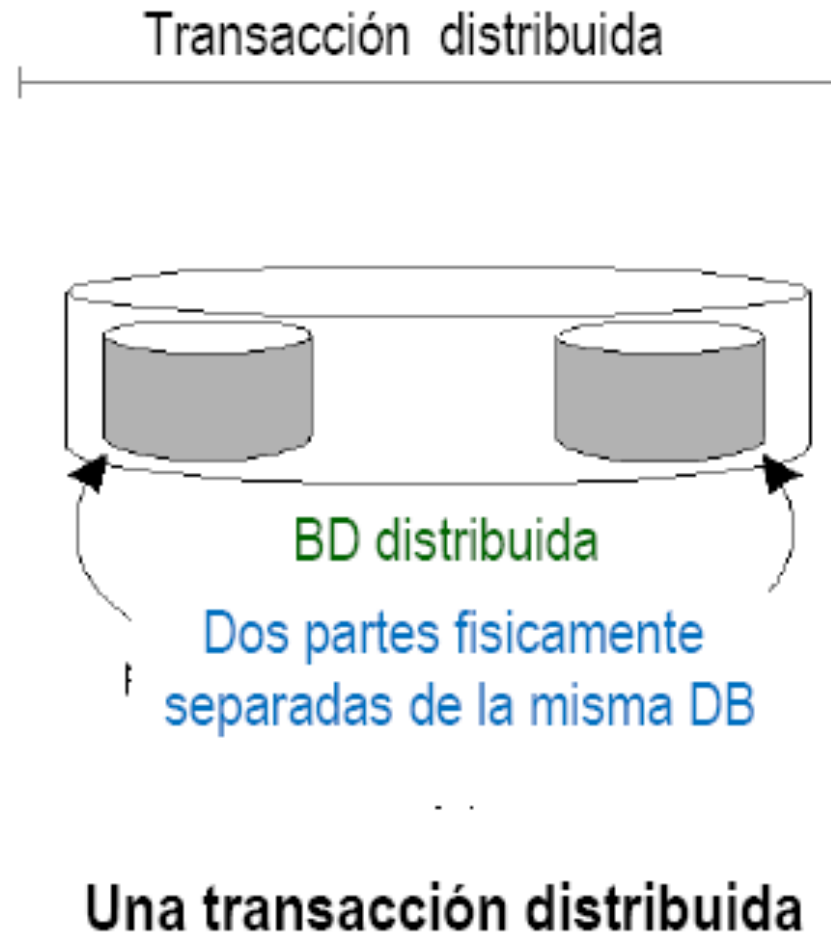
- En algunas situaciones, el cliente necesitan que una secuencia de operaciones a la base de datos distribuida se ejecuten de manera atómica:
  - libres de interferencia por operaciones de otros clientes.
  - Todas las operaciones se deben completar con éxito o no tener ningún efecto si el servidor falla.

# Transacciones

- Modelo de las transacciones



# Transacciones distribuidas





# Transacciones: Propiedades ACID

- Atomicidad (**A**tomicity)
- Consistencia (**C**onsistency)
- Aislamiento (**I**solation)
- Durabilidad (**D**urability)

# Transacciones

- Ejemplo de transacciones

```
Begin_transaction ACTUALIZAR
begin
    EXEC SQL UPDATE cuentas
              SET saldo=saldo-1000
              WHERE id_cuenta=1111
end
```

*Donde N1 y N2 son los dos nodos existentes en el Sistema de B.D.D*

```
Begin_transaction RESERVAR
begin
    EXEC SQL UPDATE N1.cuentas SET saldo=saldo-1000 WHERE id_cuenta=1111
    EXEC SQL UPDATE N2.cuentas SET saldo=saldo+1000 WHERE id_cuenta=222
end
```

# Transacciones: Condiciones de terminación

Una transacción siempre termina, aun cuando exista un fallo.

Si una transacción termina de manera exitosa se dice que la transacción ha sido consumada: ***commit*** .

El resultado de una transacción consumada es almacenado en la base de datos de manera permanente y no se puede deshacer.

# Transacciones: Condiciones de terminación

Si la transacción se detiene sin terminar su tarea, se dice que la transacción ***aborta***.

Cuando la transacción es abortada, su ejecución se detiene y todas las acciones ejecutadas hasta el momento se deshacen (*undone*), regresando la base de datos al estado antes de su ejecución.

A esta operación también se le conoce como *rollback*.

# Transacciones: Condiciones de terminación

```
Begin_transaction RESERVAR
```

```
begin
```

```
    EXEC SQL UPDATE N1.cuentas SET saldo=saldo-1000 WHERE id_cuenta=1111
```

```
    EXEC SQL UPDATE N2.cuentas SET saldo=saldo+1000 WHERE id_cuenta=222
```

```
    commit;
```

```
end
```

```
if (catcha una excepción en alguna operación) then
```

```
    rollback;
```

# Tipos de Transacciones

Clasificación de acuerdo a su estructura

- Transacciones planas: Estas transacciones tienen un punto de partida simple (**Begin\_transaction**) y un punto simple de terminación (**End\_transaction**)

```
Begin_transaction RESERVAR
```

```
begin
```

```
    EXEC SQL  UPDATE cuentas SET saldo=saldo-1000 WHERE id_cuenta=1111
```

```
    EXEC SQL  UPDATE cuentas SET saldo=saldo+1000 WHERE id_cuenta=222
```

```
end
```

# Transacciones anidadas

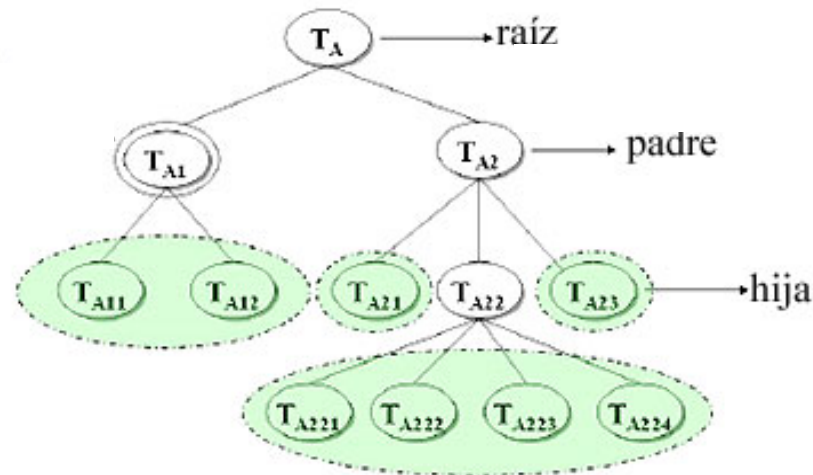
Las transacciones anidadas: las operaciones de una transacción anidada pueden incluir otras transacciones.

```
BeginTransaction Reservación
    BeginTransaction Vuelo
    ...
    EndTransaction {Vuelo}
    BeginTransaction Hotel
    ...
    endTransaction {Hotel}
    BeginTransaction Car
    ...
    endTransaction {Car}

EndTransaction {Reservación}
```

# Transacciones anidadas

- Una transacción anidada dentro de otra transacción conserva las mismas propiedades que la de sus padres, esto implica, que puede contener así mismo transacciones dentro de ella.



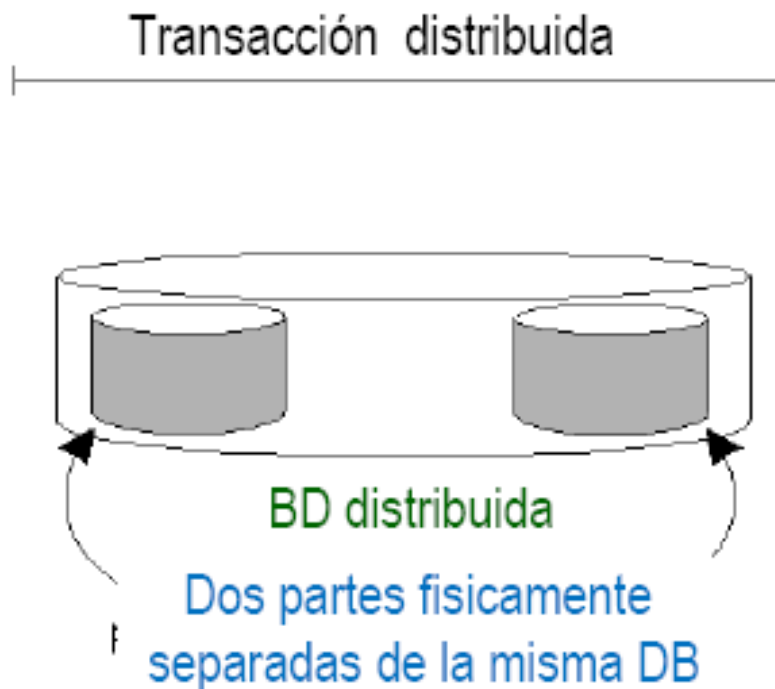


# Transacciones anidadas

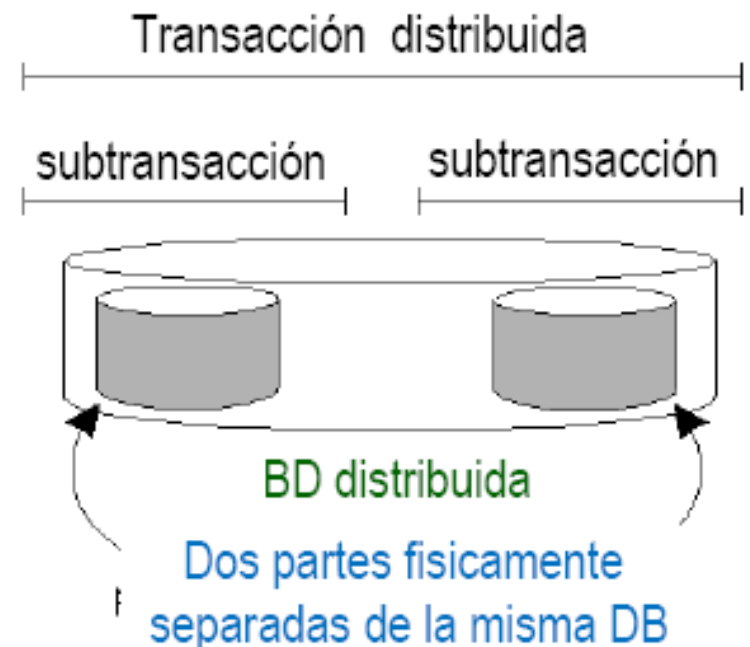
- Existen restricciones para una transacción anidada:
  - Debe empezar después que su padre y debe terminar *antes* que él.
  - El *commit* de una transacción padre está condicionada al *commit* de sus transacciones hijas
  - Si alguna transacción hija aborta (*rollback*), la transacción padre también será abortada (*rollback*).

# Transacciones subtransacciones

distribuidas con



Una transacción distribuida



Una transacción distribuida

# Transacciones: Bitácora

- Es un archivo que permite deshacer las operaciones realizadas sobre una o varias bases de datos en caso de que falle la transacción.
- Esto se hace con el fin de mantener la integridad de la información y que la transacción sea atómica.

# Transacciones: Bitácora

```
x = 0;  
y = 0;  
BEGIN_TRANSACTION;  
  x = x + 1;  
  y = y + 2  
  x = y * y;  
END_TRANSACTION;
```

Bitácora

[x = 0 / 1]

Bitácora

[x = 0 / 1]

[y = 0/2]

Bitácora

[x = 0 / 1]

[y = 0/2]

[x = 1/4]