

Modelos y bases de datos

A wide-angle, high-angle photograph of a vast data center. The space is filled with rows of server racks, each illuminated with a warm, orange glow. The racks are connected by a complex network of cables and conduits, creating a dense, organized web of infrastructure. The ceiling is high, with a visible steel truss structure and numerous long, horizontal light fixtures. The floor is a light-colored, polished surface that reflects the ambient light. The overall atmosphere is one of a highly advanced, industrial-scale computing environment.

Google Data Center



Tener en cuenta...

- Proyecto: Modelo Físico
Sábado 10 de Mayo
- Autoestudio 5
Jueves 24 de Abril – 10:00 am
- Sesión 15
Viernes 2 de Mayo – 10:00 am

Proyecto

AGENDA

S03. FORMULACIÓN DEL PROYECTO. (SA 8 FEB)

S05. DISEÑO CONCEPTUAL. GENERAL. (SA 22 FEB)

S07. DISEÑO CONCEPTUAL. EXTENDIDO. (SA 08 MAR)

S09. DISEÑO LÓGICO. ESTRUCTURA + DECLARATIVAS. (SA 22 MAR)

S11 . DISEÑO LÓGICO. ESTRUCTURA + PROCEDIMENTALES. (SA 05 ABR -> SA 12 ABR)

S15. DISEÑO FÍSICO. DATOS + COMPONENTES (SA 10 MAY)

S16. ENTREGA FINAL. PRIMER CICLO (SA 17 MAY).

S18. ENTREGA FINAL. SEGUNDO CICLO. (LU 19 MAY)

Prácticas XP

Testing

- ⓧ All code must have unit tests.
- ⓧ All code must pass all unit tests before it can be released.
- ⓧ When a bug is found tests are created.
- ⓧ Acceptance tests are run often and the score is published.

¿Qué propone?

¿Para qué se utiliza?

¿Cómo la usarían en MBDA?

Propiedades ACID

(Propiedades de una transacción)

EXAMPLE

A ATOMICIDAD	Los cambios se realizan como si fuera una sola operación (Todas o ninguna)	Si transfieres dinero entre cuentas, el débito y el crédito deben ocurrir juntos.
C CONSISTENCIA	Los datos están en un estado coherente cuando inicia una operación y cuando finaliza	Las reglas de integridad (como claves primarias o restricciones de tipo) deben mantenerse.
I AISLAMIENTO	<ul style="list-style-type: none">El estado intermedio de una transacción es invisible para otras transacciones.Las transacciones concurrentes no deben interferir entre sí	Si dos personas compran el último producto al mismo tiempo, solo una debe lograrlo .
D DURABILIDAD	Al finalizar una transacción correctamente, los cambios se persisten y no se deshacen (incluso ante fallos del sistema).	Si el sistema se apaga repentinamente, los cambios ya confirmados no se pierden .

Concurrencia

$t = \$1.000.000$

TRANSACCIÓN A	TRANSACCIÓN B
<p>SELECT t</p> <p>-----</p> <p>+ 250.000</p> <p>-----</p>	<p>SELECT t</p> <p>-----</p> <p>- 500.000</p>

- ¿Cuál transacción se ejecuta primero?
- ¿Cuál transacción se completa?
- ¿Cuál transacción se descarta?
- ¿Cuál es el nuevo valor de t ?

Bloqueos

(Alternativa de solución 1)

- **Objetivo:** Cuando una transacción quiere asegurar que el objeto en el que está interesada (normalmente una tupla) no cambiará mientras la está usando adquiere un bloqueo sobre ese objeto.

Tipos de bloqueo

- **Exclusivo (x):** Si la transacción **A** pone un bloqueo exclusivo X sobre la tupla t, se rechazará la petición de cualquier otra transacción **B** para un bloqueo de cualquier tipo sobre t.
- **Compartido (s):** Si la transacción **A** pone un bloqueo compartido S sobre la tupla t, se rechazará la petición de cualquier otra transacción B para un bloqueo **X** sobre t y se otorgará una petición de cualquier otra transacción para un bloqueo S sobre t.

Bloqueos

(Alternativa de solución 1)

EXCLUSIVO (X)	COMPARTIDO (S)
<ul style="list-style-type: none">• Permite leer y escribir el dato.• Solo una transacción puede tener este bloqueo.• No permite ni lecturas ni escrituras por otros.	<ul style="list-style-type: none">• Permite leer un dato, pero no modificarlo.• Varias transacciones pueden tener un bloqueo compartido al mismo tiempo.• No permite escritura mientras exista.

Bloqueos

(Alternativa de solución 1)

	X	S	—	Tiene
X				
S				

Solicita

X: Exclusivo

S: Compartido

Bloqueos

(Alternativa de solución 1)

	X	S	—	Tiene
X	NO	NO	SI	
S	NO	SI	SI	

Solicita

X: Exclusivo

S: Compartido

Niveles de aislamiento



READ UNCOMMITTED

- Lectura no registrada
- Nivel de aislamiento más bajo en bases de datos relacionales.
- Permite que una transacción lea datos que aún no han sido confirmados (*commit*) por otras transacciones.

Lectura sucia

READ COMMITTED

- Lectura registrada
- Garantiza que una transacción solo puede leer datos que ya han sido confirmados (*commit*) por otras transacciones.

Lecturas no
repetibles

REPEATABLE READ

- Lectura repetible.
- Garantiza que si una transacción lee un dato, ese mismo dato no podrá ser modificado por otras transacciones hasta que finalice

Lecturas
fantasma

SERIALIZABLE

- Serializable.
- Nivel de aislamiento más estricto.
- Garantiza que el resultado de la ejecución de transacciones concurrentes sea el mismo que si se hubieran ejecutado una tras otra (en serie), sin superposición.
- Bloqueo exclusivo para actualizar y compartido para leer.
- Bloqueos de rango en las consultas (no cambiar WHERE)

Bloqueo mortal

Niveles de aislamiento



READ UNCOMMITTED

- Lectura no registrada
- Nivel de aislamiento más bajo en bases de datos relacionales.
- Permite que una transacción lea datos que aún no han sido confirmados (*commit*) por otras transacciones.

Lectura sucia

READ COMMITTED

- Lectura registrada
- Garantiza que una transacción solo puede leer datos que ya han sido confirmados (*commit*) por otras transacciones.

Lecturas no repetibles

REPEATABLE READ

- Lectura repetible.
- Garantiza que si una transacción lee un dato, ese mismo dato no podrá ser modificado por otras transacciones hasta que finalice

Lecturas fantasma

SERIALIZABLE

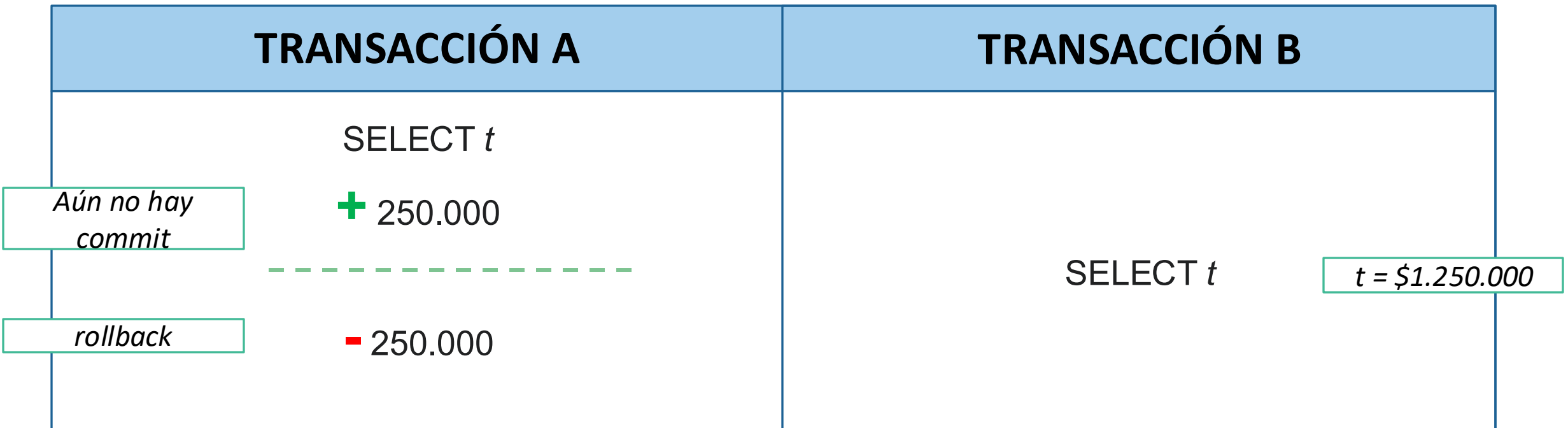
- Serializable.
- Nivel de aislamiento más estricto.
- Garantiza que el resultado de la ejecución de transacciones concurrentes sea el mismo que si se hubieran ejecutado una tras otra (en serie), sin superposición.
- Bloqueo exclusivo para actualizar y compartido para leer.
- Bloqueos de rango en las consultas (no cambiar WHERE)

Bloqueo mortal

Dependencia no confirmada

(Problema: Lectura sucia)

$t = \$1.000.000$



- ¿Cuál transacción se ejecuta primero?
- ¿Cuál transacción se completa?
- ¿Cuál es el valor de t ?

Niveles de aislamiento



READ UNCOMMITTED

- Lectura no registrada
- Nivel de aislamiento más bajo en bases de datos relacionales.
- Permite que una transacción lea datos que aún no han sido confirmados (*commit*) por otras transacciones.

Lectura sucia

READ COMMITTED

- Lectura registrada
- Garantiza que una transacción solo puede leer datos que ya han sido confirmados (*commit*) por otras transacciones.

Lecturas no repetibles

REPEATABLE READ

- Lectura repetible.
- Garantiza que si una transacción lee un dato, ese mismo dato no podrá ser modificado por otras transacciones hasta que finalice

Lecturas fantasma

SERIALIZABLE

- Serializable.
- Nivel de aislamiento más estricto.
- Garantiza que el resultado de la ejecución de transacciones concurrentes sea el mismo que si se hubieran ejecutado una tras otra (en serie), sin superposición.
- Bloqueo exclusivo para actualizar y compartido para leer.
- Bloqueos de rango en las consultas (no cambiar WHERE)

Bloqueo mortal

Actualización perdida

(Problema: Lectura no repetible)

Ocurre cuando dos o más transacciones **leen y modifican los mismos datos al mismo tiempo**, y **una de las actualizaciones se pierde porque es sobrescrita por otra**.

TRANSACCIÓN A	TRANSACCIÓN B
<p>SELECT t</p> <p>-----</p> <p>+ 250.000</p> <p>-----</p>	<p>SELECT t</p> <p>-----</p> <p>- 500.000</p>

$t = \$1.000.000$

- ¿Cuál es el valor de t ?

Actualización perdida

(Problema: Lectura no repetible)

Ocurre cuando dos o más transacciones **leen y modifican los mismos datos al mismo tiempo**, y **una de las actualizaciones se pierde porque es sobrescrita por otra**.

TRANSACCIÓN A	TRANSACCIÓN B
<p>SELECT t</p> <p>-----</p> <p>+ 250.000</p> <p>-----</p>	<p>SELECT t</p> <p>-----</p> <p>- 500.000</p>

$t = \$1.000.000$

- ¿Cuál es el valor de t ? \$500.000

Actualización perdida

(Problema: Lectura no repetible)



¿Por qué es un problema?

- **Inconsistencia de datos:** El estado final de la base de datos no refleja correctamente las operaciones realizadas.
- **Violación de integridad:** Puede generar errores graves en aplicaciones críticas como banca, inventarios, etc.



¿Cómo se previene?

- Uso de **bloqueos** para evitar accesos simultáneos.
- **Niveles de aislamiento.**
- Técnicas de control de concurrencia ***timestamp ordering o multiversion concurrency control (MVCC).***

Actualización perdida

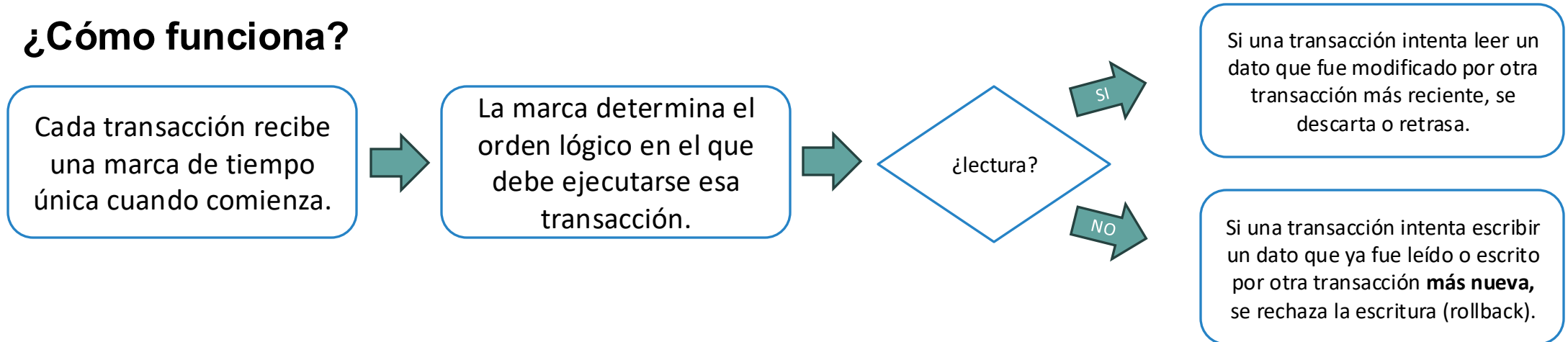
(Problema: Lectura no repetible)



Timestamp Ordering (*Ordenamiento por marcas de tiempo*)

- **Objetivo:** Asegurar que las transacciones se ejecuten de forma **serializable**, es decir, como si se ejecutaran una tras otra, aunque en realidad se hagan en paralelo.

¿Cómo funciona?



Actualización perdida

(Problema: Lectura no repetible)

Timestamp = 10 TRANSACCIÓN A	Timestamp = 20 TRANSACCIÓN B
<p data-bbox="563 582 805 629">SELECT t</p> <p data-bbox="428 691 996 705">-----</p> <p data-bbox="555 782 810 836">+ 250.000</p> <p data-bbox="448 925 1014 939">-----</p>	<p data-bbox="1691 672 1933 719">SELECT t</p> <p data-bbox="1556 796 2125 811">-----</p> <p data-bbox="1691 896 1926 943">- 500.000</p>

$t = \$1.000.000$

- ¿Cuál transacción se ejecuta primero?
- ¿Cuál transacción se completa?
- ¿Cuál transacción se descarta?

Actualización perdida

(Problema: Lectura no repetible)

<div>Timestamp = 10</div> <div>TRANSACCIÓN A</div>	<div>Timestamp = 20</div> <div>TRANSACCIÓN B</div>
<div>SELECT t</div> <div>-----</div> <div>+ 250.000</div> <div>-----</div>	<div>SELECT t</div> <div>-----</div> <div>- 500.000</div>

$t = \$1.000.000$

- ¿Cuál transacción se ejecuta primero? Transacción A
- ¿Cuál transacción se completa? Transacción B
- ¿Cuál transacción se descarta? Transacción A

Actualización perdida

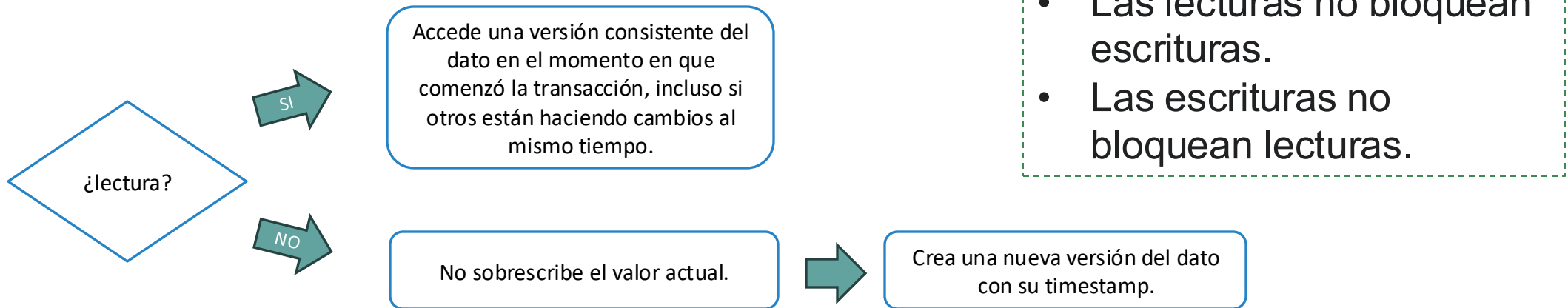
(Problema: Lectura no repetible)



Multiversion Concurrency Control (MVCC)

- **Objetivo:** Permitir que varias transacciones accedan simultáneamente a los mismos datos **sin bloquearse entre sí**. En lugar de bloquear los datos, MVCC mantiene múltiples versiones de los mismos registros.

¿Cómo funciona?



Actualización perdida

$t = \$1.000.000$

(Problema: Lectura no repetible)

<div>Timestamp = 10</div> <div>TRANSACCIÓN A</div>	<div>Timestamp = 20</div> <div>TRANSACCIÓN B</div>
<div>SELECT t</div> <div>-----</div> <div>+ 250.000</div> <div>-----</div>	<div>SELECT t</div> <div>-----</div> <div>- 500.000</div>

- ¿Cuál transacción se ejecuta primero?
- ¿Cuál transacción se completa?
- ¿Cuál transacción se descarta?

Actualización perdida

$t = \$1.000.000$

(Problema: Lectura no repetible)

<div>Timestamp = 10</div> <div>TRANSACCIÓN A</div>	<div>Timestamp = 20</div> <div>TRANSACCIÓN B</div>
<div data-bbox="563 582 805 629">SELECT t</div> <div data-bbox="425 694 996 708">-----</div> <div data-bbox="555 782 810 836"><div>+</div>250.000</div> <div data-bbox="445 929 1016 943">-----</div>	<div data-bbox="1691 675 1933 722">SELECT t</div> <div data-bbox="1559 801 2130 815">-----</div> <div data-bbox="1691 896 1926 943"><div>-</div>500.000</div>
<div data-bbox="537 1075 863 1122">$t = \\$1.250.000$</div>	<div data-bbox="1671 1075 1997 1122">$t = \\$1.000.000$</div>

- ¿Cuál transacción se ejecuta primero? Transacción A
- ¿Cuál transacción se completa? Transacción A y B
- ¿Cuál transacción se descarta? Ninguna

Actualización perdida

$t = \$1.000.000$

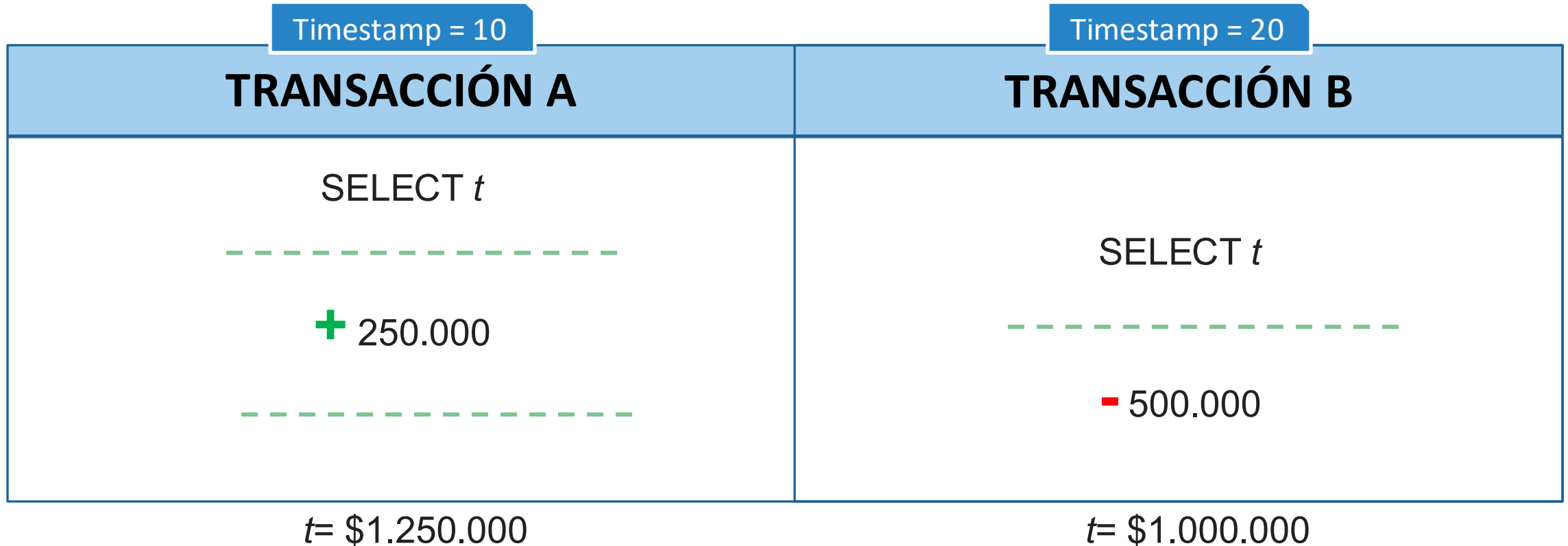
(Problema: Lectura no repetible)

<div>Timestamp = 10</div> <div>TRANSACCIÓN A</div>	<div>Timestamp = 20</div> <div>TRANSACCIÓN B</div>
<div data-bbox="563 582 805 629">SELECT t</div> <div data-bbox="555 782 812 836"><div>+</div>250.000</div>	<div data-bbox="1691 675 1933 722">SELECT t</div> <div data-bbox="1691 896 1928 943"><div>-</div>500.000</div>

Actualización perdida

$t = \$1.000.000$

(Problema: Lectura no repetible)



- ¿Cuál transacción se ejecuta primero? Transacción A
- ¿Cuál transacción se completa? Transacción A
- ¿Cuál transacción se descarta? Transacción B

Niveles de aislamiento



READ UNCOMMITTED

- Lectura no registrada
- Nivel de aislamiento más bajo en bases de datos relacionales.
- Permite que una transacción lea datos que aún no han sido confirmados (*commit*) por otras transacciones.

Lectura sucia

READ COMMITTED

- Lectura registrada
- Garantiza que una transacción solo puede leer datos que ya han sido confirmados (*commit*) por otras transacciones.

Lecturas no repetibles

REPEATABLE READ

- Lectura repetible.
- Garantiza que si una transacción lee un dato, ese mismo dato no podrá ser modificado por otras transacciones hasta que finalice

Lecturas fantasma

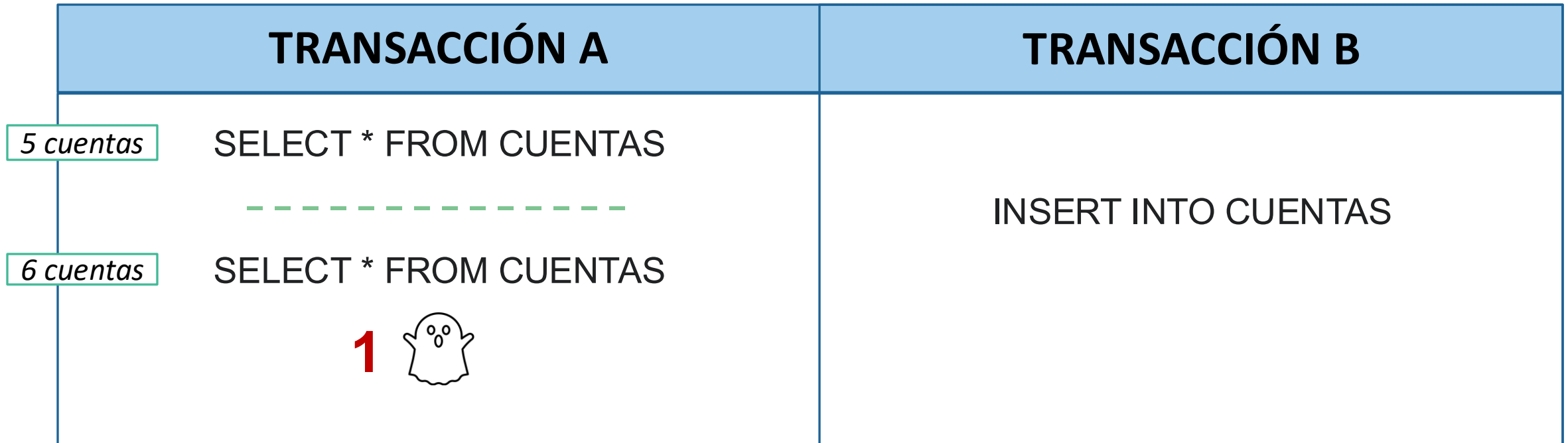
SERIALIZABLE

- Serializable.
- Nivel de aislamiento más estricto.
- Garantiza que el resultado de la ejecución de transacciones concurrentes sea el mismo que si se hubieran ejecutado una tras otra (en serie), sin superposición.
- Bloqueo exclusivo para actualizar y compartido para leer.
- Bloqueos de rango en las consultas (no cambiar WHERE)

Bloqueo mortal

Problema: Lectura fantasma

Cuando una transacción ejecuta una consulta que devuelve un conjunto de filas, y más tarde la misma consulta devuelve más filas porque otra transacción insertó nuevos registros.

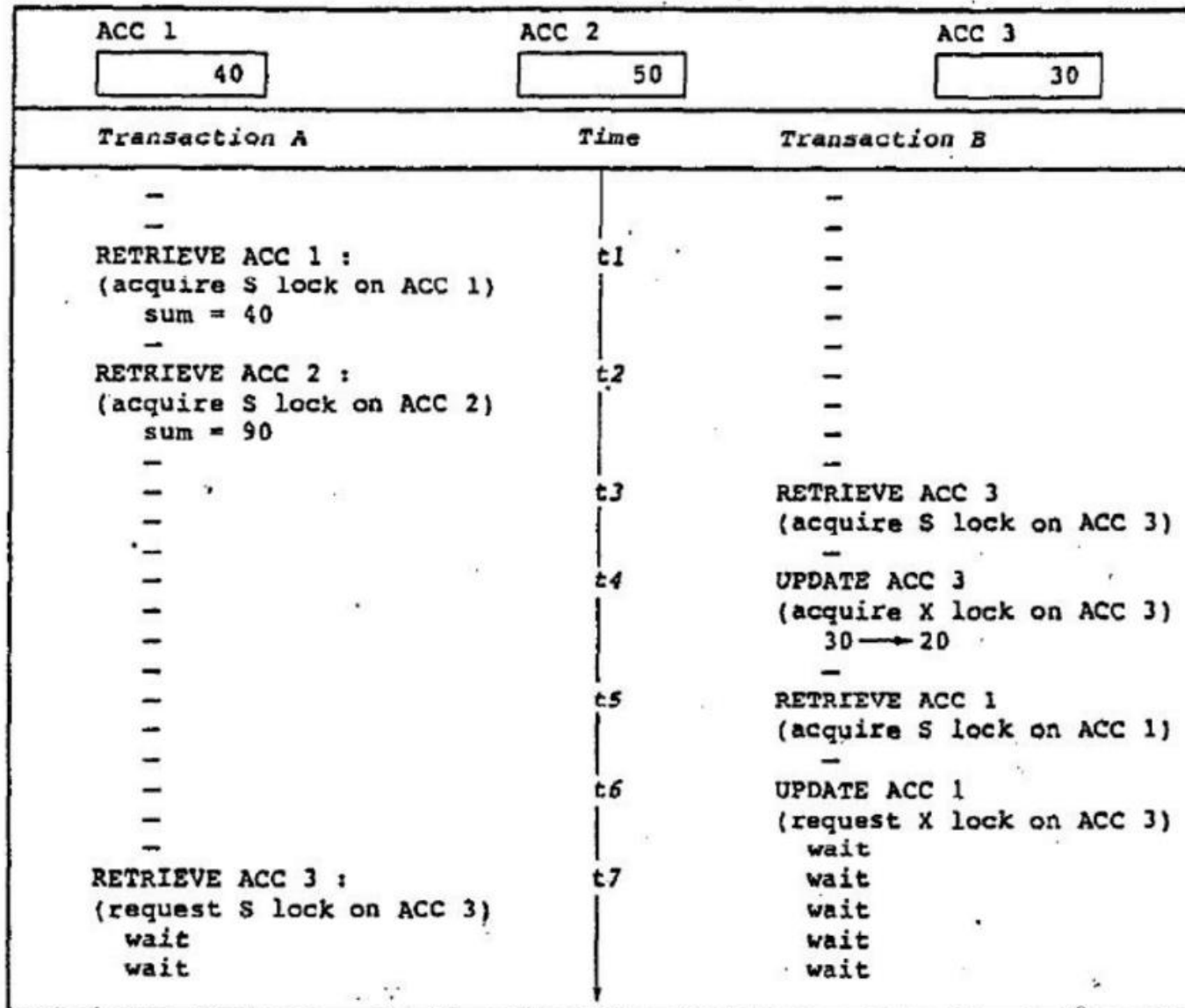


Niveles de aislamiento



READ UNCOMMITTED	<ul style="list-style-type: none">• Lectura no registrada• Nivel de aislamiento más bajo en bases de datos relacionales.• Permite que una transacción lea datos que aún no han sido confirmados (<i>commit</i>) por otras transacciones.	Lectura sucia
READ COMMITTED	<ul style="list-style-type: none">• Lectura registrada• Garantiza que una transacción solo puede leer datos que ya han sido confirmados (<i>commit</i>) por otras transacciones.	Lecturas no repetibles
REPEATABLE READ	<ul style="list-style-type: none">• Lectura repetible.• Garantiza que si una transacción lee un dato, ese mismo dato no podrá ser modificado por otras transacciones hasta que finalice	Lecturas fantasma
SERIALIZABLE	<ul style="list-style-type: none">• Serializable.• Nivel de aislamiento más estricto.• Garantiza que el resultado de la ejecución de transacciones concurrentes sea el mismo que si se hubieran ejecutado una tras otra (en serie), sin superposición.• Bloqueo exclusivo para actualizar y compartido para leer.• Bloqueos de rango en las consultas (no cambiar WHERE)	Bloqueo mortal

Problema: Bloqueo mortal



- ¿Cuál transacción se ejecuta primero?
- ¿Cuál transacción se completa?