Modelos y bases de datos

Recuperación y concurrencia

CEIS

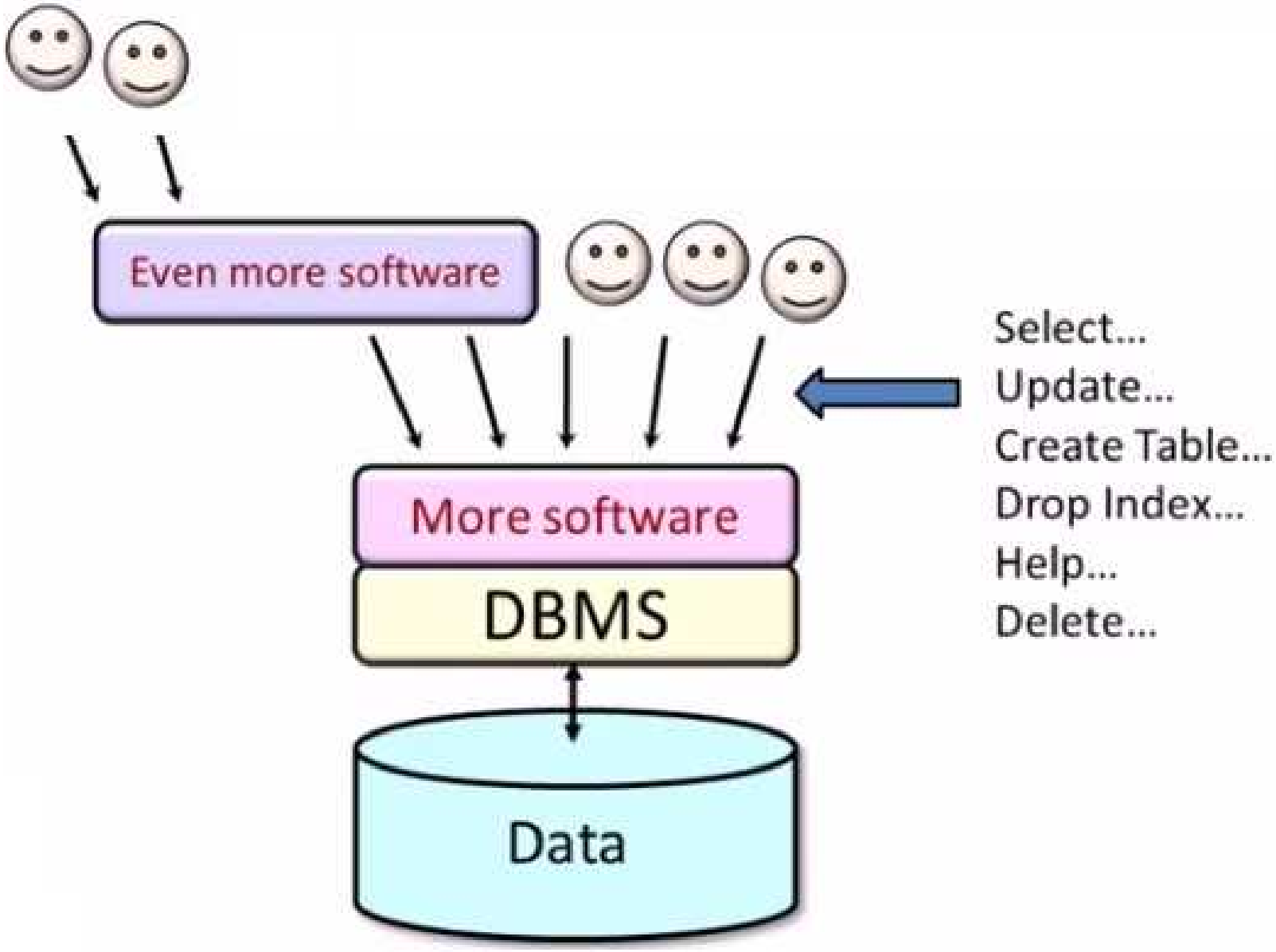
2023-2

Agenda

Agenda

Agenda

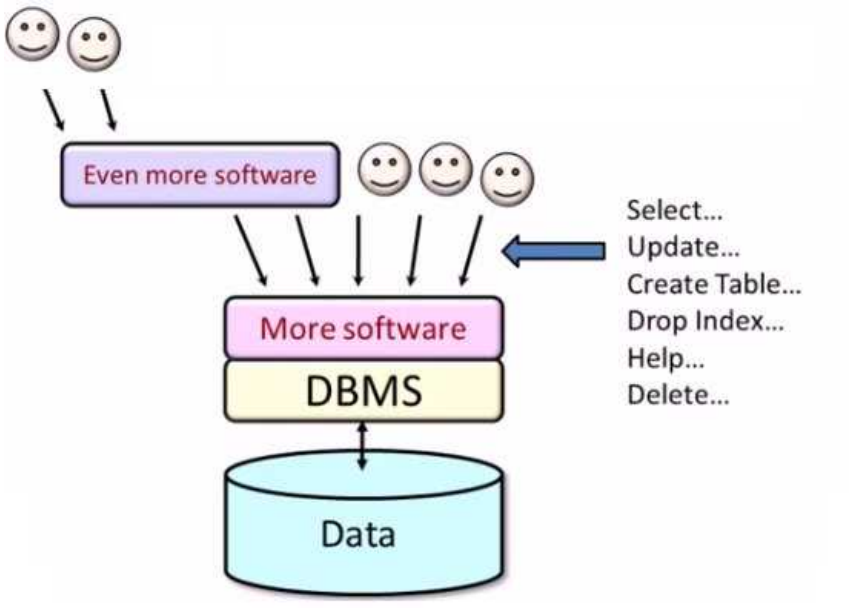
Problemas-Solución



"MOOC Databases. Standford."

Problemas-Solución

:(Dos aplicaciones que están ejecutando simultáneamente usan los mismos datos

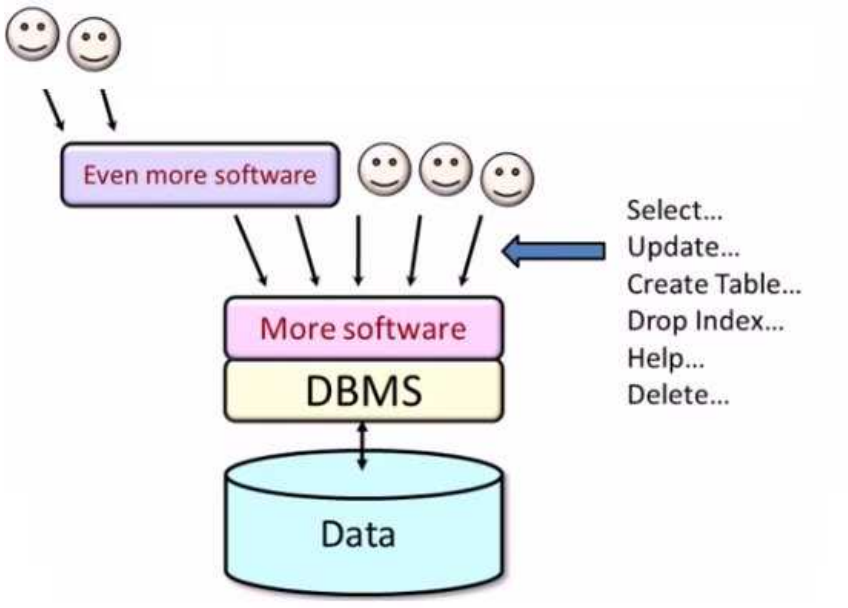


"MOOC Databases. Standford."

:(El sistema falla cuando se está ejecutando

Retirar dinero en un cajero

Problemas-Solución



"MOOC Databases. Standford."

:(Dos aplicaciones que están ejecutando simultáneamente usan los mismos datos

Puede quedar la base de datos con datos incorrectos

Concurrencia

¿Cómo garantizamos una ejecución concurrente **correcta**?

:(El sistema falla cuando se está ejecutando

Puede quedar la base de datos en un estado desconocido (correcto o incorrecto)

Recuperación

¿Cómo volvemos a un estado **correcto** y conocido?

ADMINISTRACION DE TRANSACCIONES

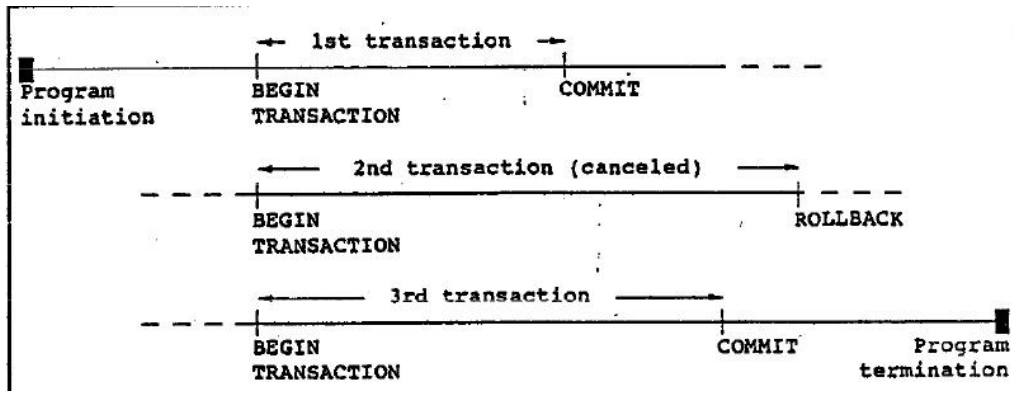
Retirar dinero en un cajero

Agenda

Transacciones

Definición

Contexto



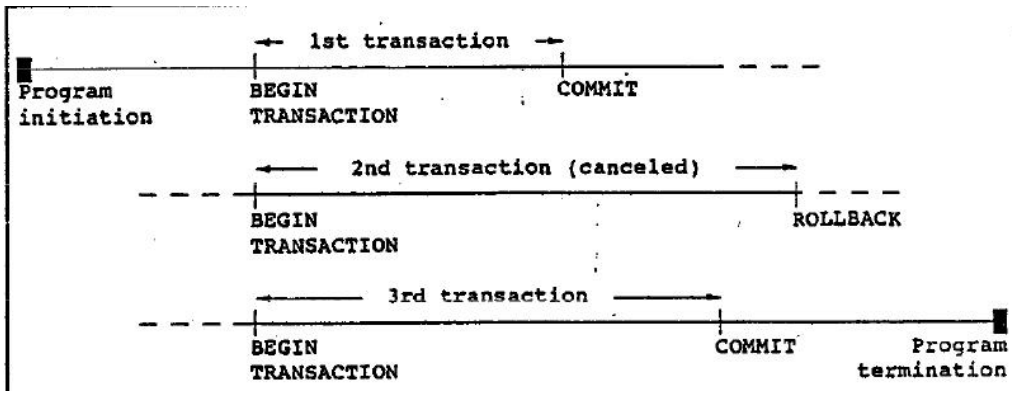
Propiedades. ACID

Transacciones

Definición

Es una secuencia de operaciones que transforman un estado consistente en otro estado consistente. Unidad de trabajo lógica.

Contexto



La ejecución de un programa es la ejecución de una secuencia de transacciones que pueden ser **concurrentes**

Propiedades. ACID

- A Atomicidad
- C Consistencia
- I Aislamiento
- D Durabilidad

Transacciones

Transferencia

```
CREATE OR REPLACE
PROCEDURE TRANSFERENCIA(origen IN  CHAR, destino IN CHAR, valor IN NUMBER)
IS
BEGIN
  UPDATE CUENTAS SET saldo=saldo+valor WHERE numero=destino;
  UPDATE CUENTAS SET saldo=saldo-valor WHERE numero=origen;
END;
/
```

Hay un chequeo de saldo$>=$0

- ¿Es correcto?

Transacciones

Transferencia

```
CREATE OR REPLACE
PROCEDURE TRANSFERENCIA(origen IN  CHAR, destino IN CHAR, valor IN NUMBER)
IS
BEGIN
  UPDATE CUENTAS SET saldo=saldo+valor WHERE numero=destino;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta destino');
  END IF;
  UPDATE CUENTAS SET saldo=saldo-valor WHERE numero=origen;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta origen');
  END IF;
END;
/
```

Hay un chequeo de saldo$>=$0

- ¿Es correcto?
- ¿Qué pasa si el sistema se cae después de realizarse la primera instrucción?

Transacciones

Transferencia

```
CREATE OR REPLACE
PROCEDURE TRANSFERENCIA(origen IN  CHAR, destino IN CHAR, valor IN NUMBER)
IS
BEGIN
  UPDATE CUENTAS SET saldo=saldo+valor WHERE numero=destino;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta destino');
  END IF;
  UPDATE CUENTAS SET saldo=saldo-valor WHERE numero=origen;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta origen');
  END IF;
END;
/
```

leer,actualizar,escribir

leer,actualizar,escribir

Hay un chequeo de saldo$>=$0

- ¿Es correcto?
- ¿Qué pasa si el sistema se cae después de realizarse la primera instrucción?
- ¿Qué puede pasar si hay dos transacciones concurrentes?

Transacciones

Elementos

UN INICIO

1. `BEGIN TRANSACTION`

   Indica el inicio de la transacción

   DOS POSIBLES FINALES

2. `COMMIT`

   Indica la finalización de una transacción satisfactoria

   Los cambios ahora son permanentes y visibles a todos los usuarios

3. `ROLLBACK`

   Indica la finalización de una transacción no satisfactoria

   Los cambios no son válidos, nadie los verá

Puntos

- **De sincronización** Las instrucciones COMMIT y ROLLBACK establecen puntos de sincronización; son puntos en que la base de datos debería estar en un estado de consistencia
- **De verificación** Escritura a disco, a determinados intervalos de tiempos prescritos

Transacciones

Transferencia

```
CREATE OR REPLACE
PROCEDURE TRANSFERENCIA(origen IN  CHAR, destino IN CHAR, valor IN NUMBER)
IS
BEGIN
  UPDATE CUENTAS SET saldo=saldo+valor WHERE numero=destino;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta destino');
  END IF;
  UPDATE CUENTAS SET saldo=saldo-valor WHERE numero=origen;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta origen');
  END IF;
END;
/
```

- ¿BEGIN TRANSACTION?
- ¿COMMIT?
- ¿ROLLBACK?

Transacciones

Transferencia

```
CREATE OR REPLACE
PROCEDURE TRANSFERENCIA(origen IN  CHAR, destino IN CHAR, valor IN NUMBER)
IS
BEGIN
  --BEGIN TRANSACTION
  UPDATE CUENTAS SET saldo=saldo+valor WHERE numero=destino;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta destino');
  END IF;
  UPDATE CUENTAS SET saldo=saldo-valor WHERE numero=origen;
  IF (SQL%ROWCOUNT=0) THEN
     RAISE_APPLICATION_ERROR(-20001,'No existe la cuenta origen');
  END IF;
  COMMIT;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    RAISE_APPLICATION_ERROR(-20999,SQLERRM);
END;
/
```

Agenda

Corrección

De un programa

De base de dase de datos

De conjunto de transacciones

Corrección

De un programa

:) Si cumple la precondición debe cumplir la poscondición

De base de dase de datos

:) Si cumple todas las restricciones de integridad definidas

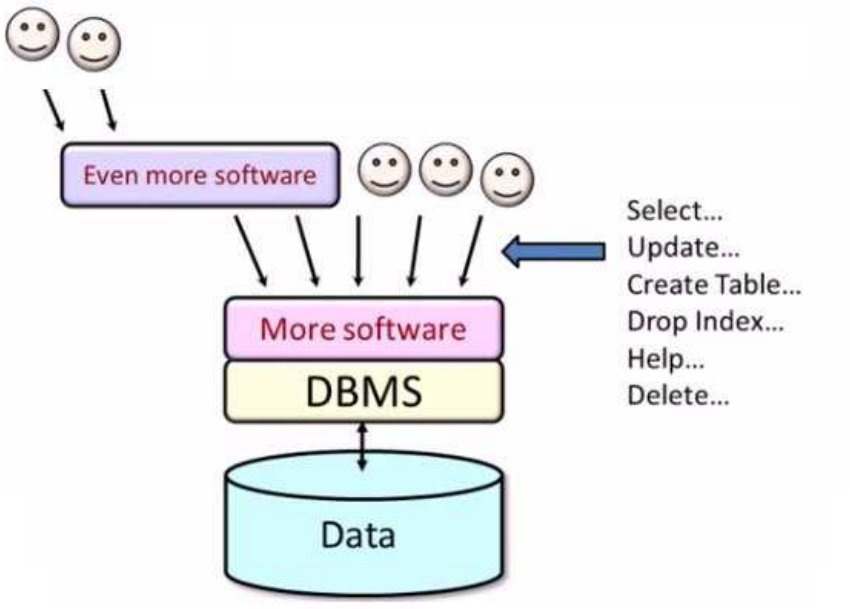
De conjunto de transacciones

:) Si es serializable

Produce el mismo resultado que **una** ejecución serial de las mismas transacciones

Agenda

Problemas-Solución



"MOOC Databases. Standford."

:(Dos aplicaciones que están ejecutando simultáneamente usan los mismos datos

Puede quedar la base de datos con datos incorrectos

Concurrencia

¿Cómo garantizamos una ejecución concurrente **correcta**?

Agenda

Problemas

1. Actualización perdida

   Lectura no repetible

2. Dependencia no confirmada

   Lectura sucia

3. Análisis inconsistente

   Lectura no repetible

4.

   Fantasmas

Problemas

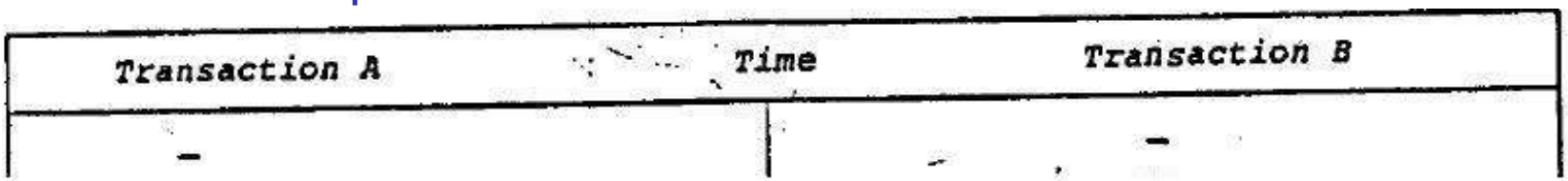
Actualización perdida

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Cuáles serían posibles resultados de una ejecución correcta?

Problemas

Actualización perdida

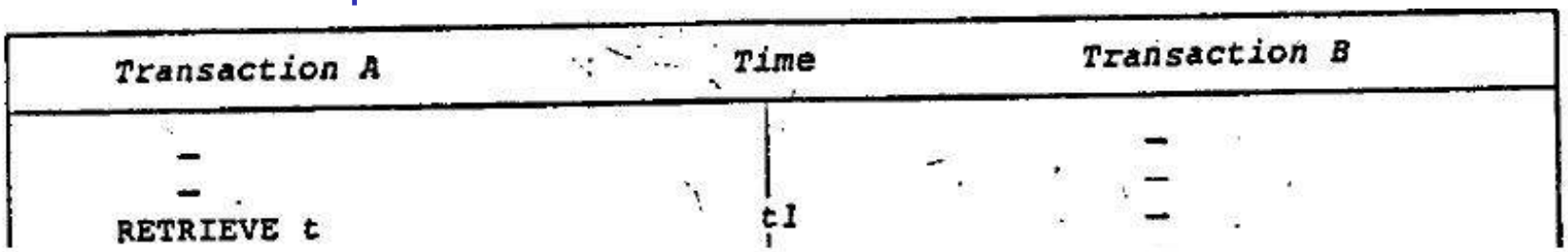


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Problemas

Actualización perdida

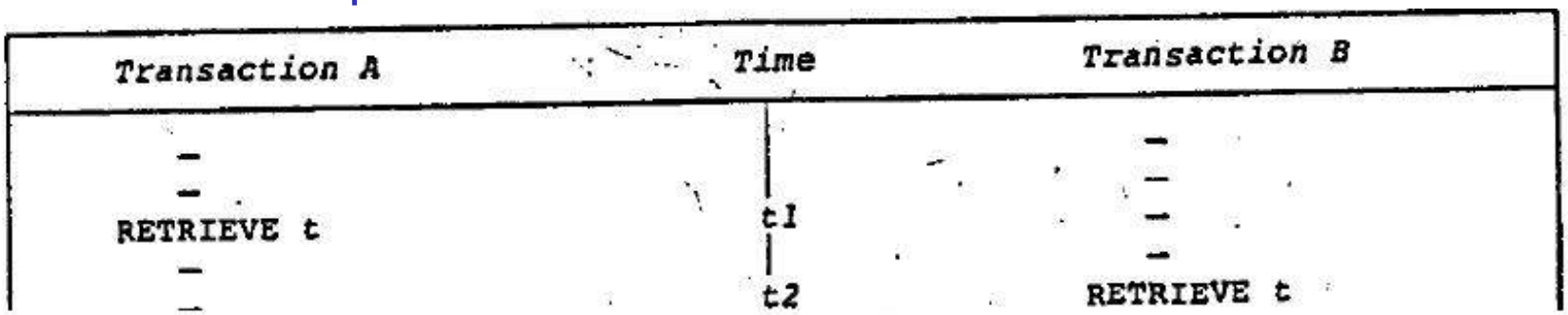


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Problemas

Actualización perdida

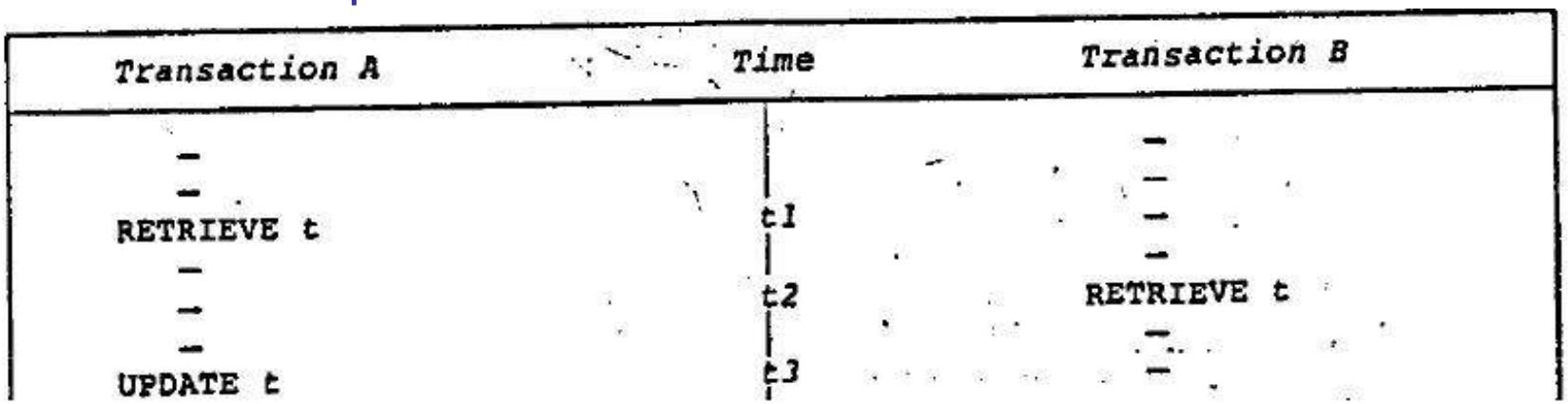


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Problemas

Actualización perdida

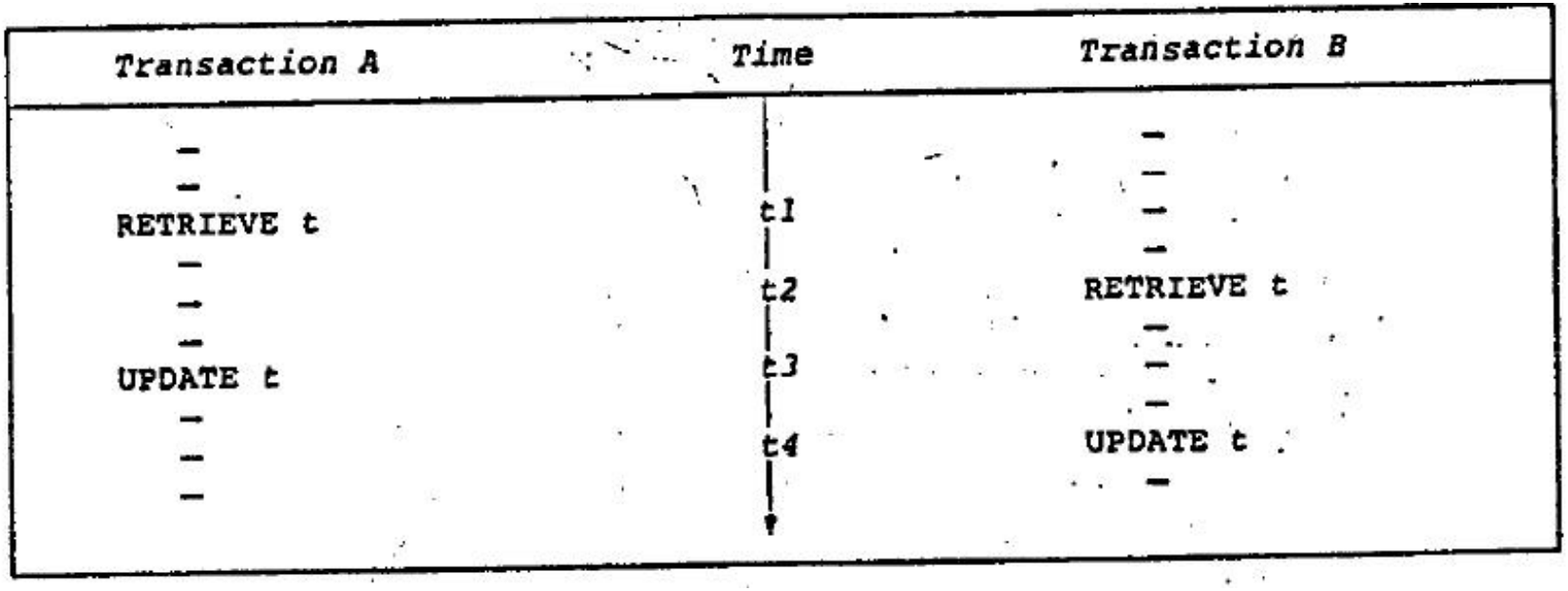


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Problemas

Actualización perdida

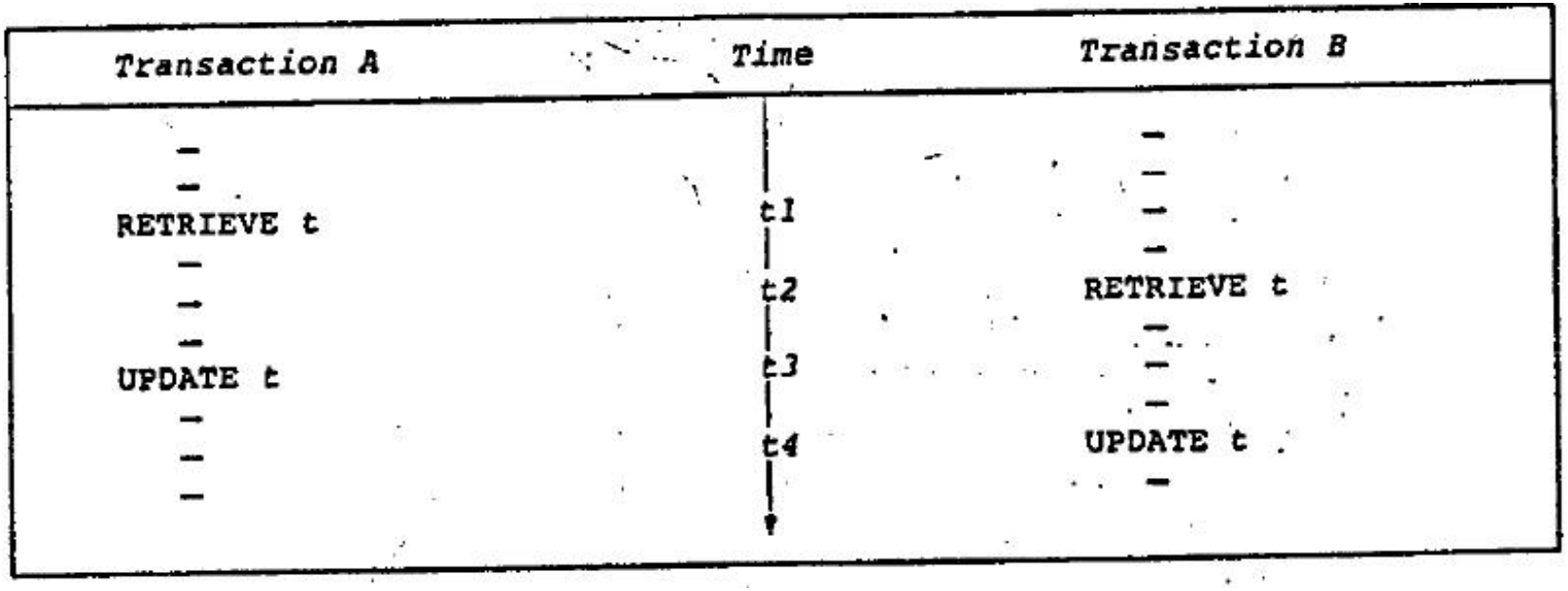


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Problemas

Actualización perdida

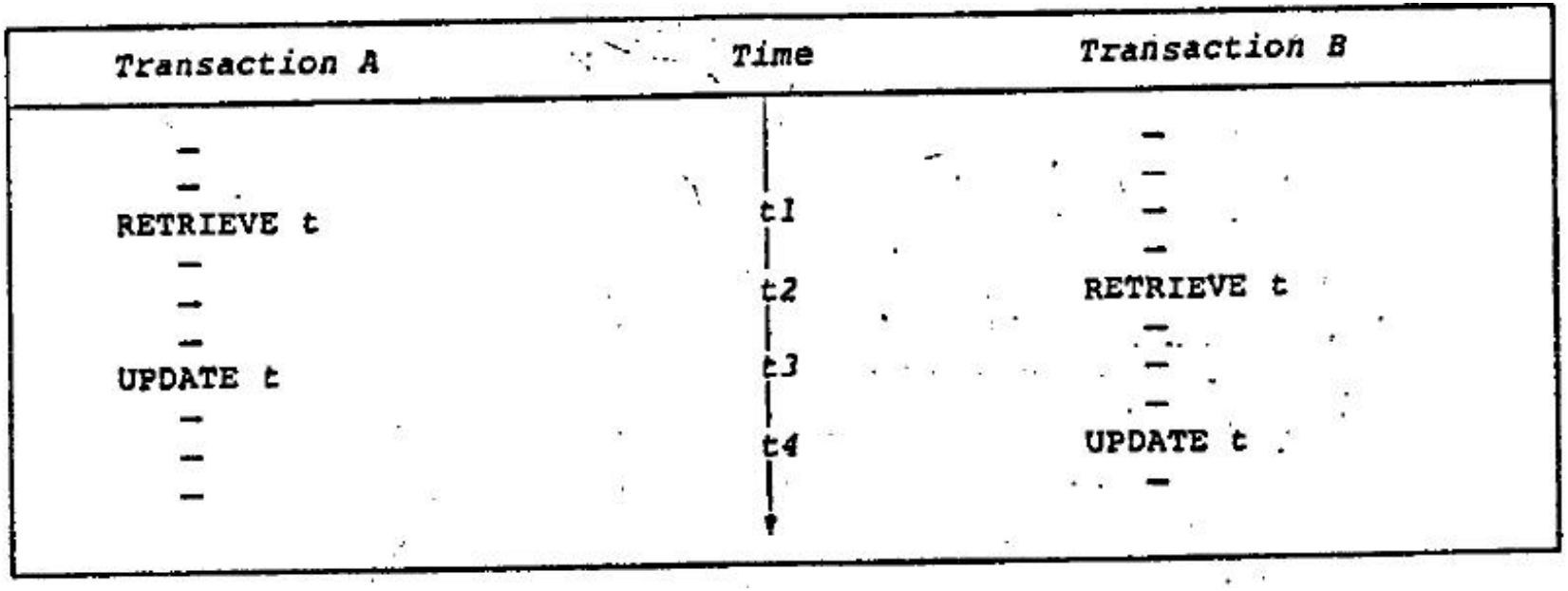


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Es correcto?

Problemas

Actualización perdida



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Es correcto? ¡No es correcto! Lectura no repetible

Agenda

Bloqueo

Idea

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)
- Compartido (S)

Bloqueo

Idea

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`

  – se otorgará una petición de cualquier otra transacción para un bloqueo `S` sobre `t`

Bloqueo

Idea

Cuando una transacción quiere asegurar que 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)
- Compartido (S)

Completar tabla

	X	S	-
X			
S			

Tlene

Solicita

Bloqueo

Idea

Cuando una transacción quiere asegurar que objeto en el que está interesada no cambiará mientras la está usando adquiere un **bloqueo** sobre ese objeto.

Granularidad del bloqueo

- Tupla

  Normalmente es una tupla
- Tabla
- Base de datos

Transacciones

Propiedadades. ACID. [I] Aislamiento. Niveles.

1. READ UNCOMMITTED : Lectura no registrada
2. READ COMMITTED : Lectura registrada
3. REPEABLE READ : Lectura repetible
4. SERIALIZABLE : Serializable

Transacciones

Propiedadades. ACID. [I] Aislamiento. Niveles.

1. READ UNCOMMITTED : Lectura no registrada

   sin bloqueos
2. READ COMMITTED : Lectura registrada

   Bloqueo exclusivo para actualizar
3. REPEABLE READ : Lectura repetible

   Bloqueo exclusivo para actualizar y compartido para leer
4. SERIALIZABLE : Serializable

   Bloqueo exclusivo para actualizar y compartido para leer

   Bloqueos de rango en las consultas (no cambiar WHERE)

Bloqueo

Protocolo: Lectura repetible

Este es el protocolo normal de acceso a los datos

Operaciones - bloqueo

- Para leer

  Una transacción que desea recuperar una tupla primero debe adquirir un bloqueo compartido S sobre la tupla

  (S) Compartido = De lectura

- Para escribir

  Una transacción que desea actualizar una tupla primero debe adquirir un bloqueo exclusivo X sobre la tupla

  (X) Exclusivo = De escritura

Con bloqueos: Lectura repetible

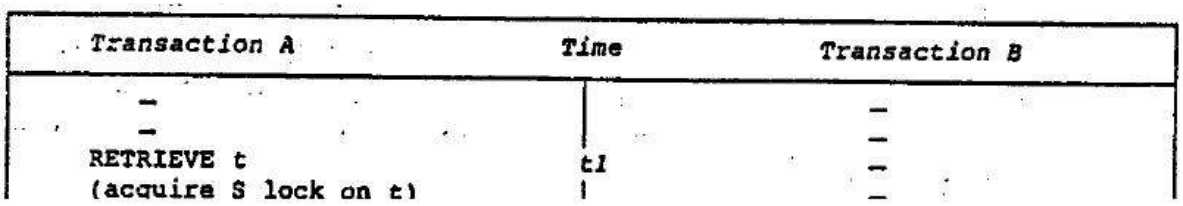
Actualización perdida

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Cuáles serían posibles resultados de una ejecución correcta?

Con bloqueos: Lectura repetible

Actualización perdida



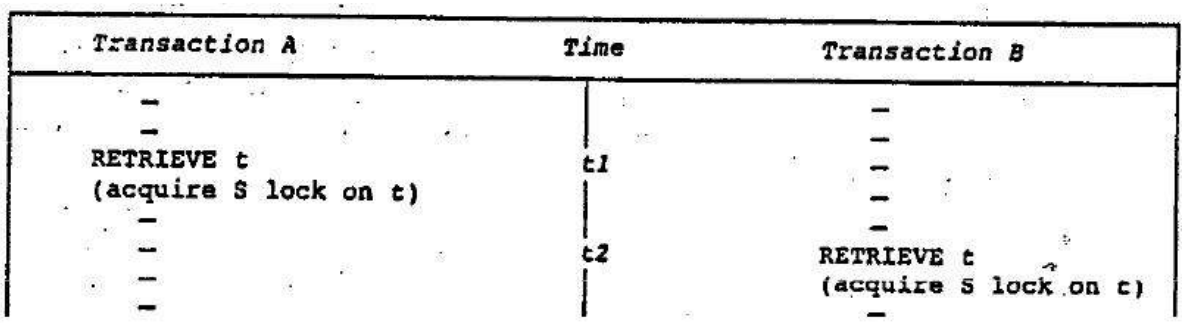
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)	|	–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Con bloqueos: Lectura repetible

Actualización perdida



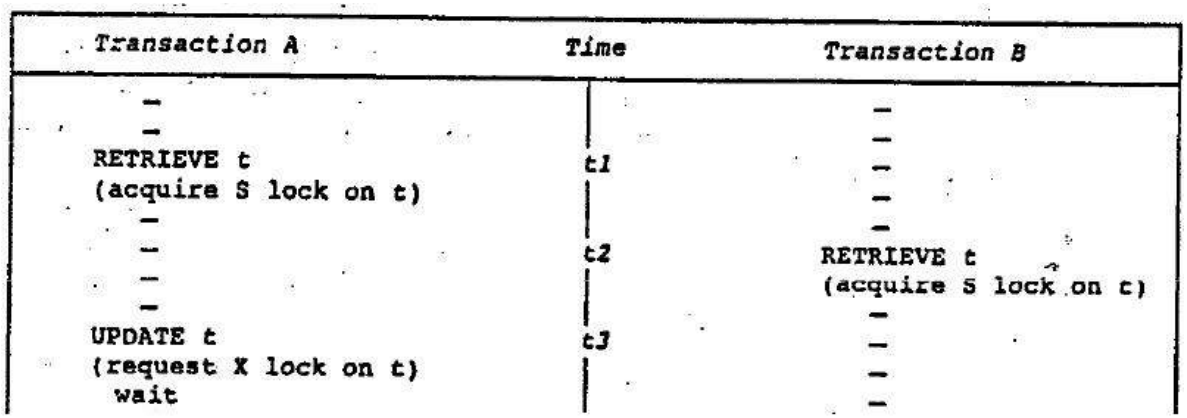
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Con bloqueos: Lectura repetible

Actualización perdida



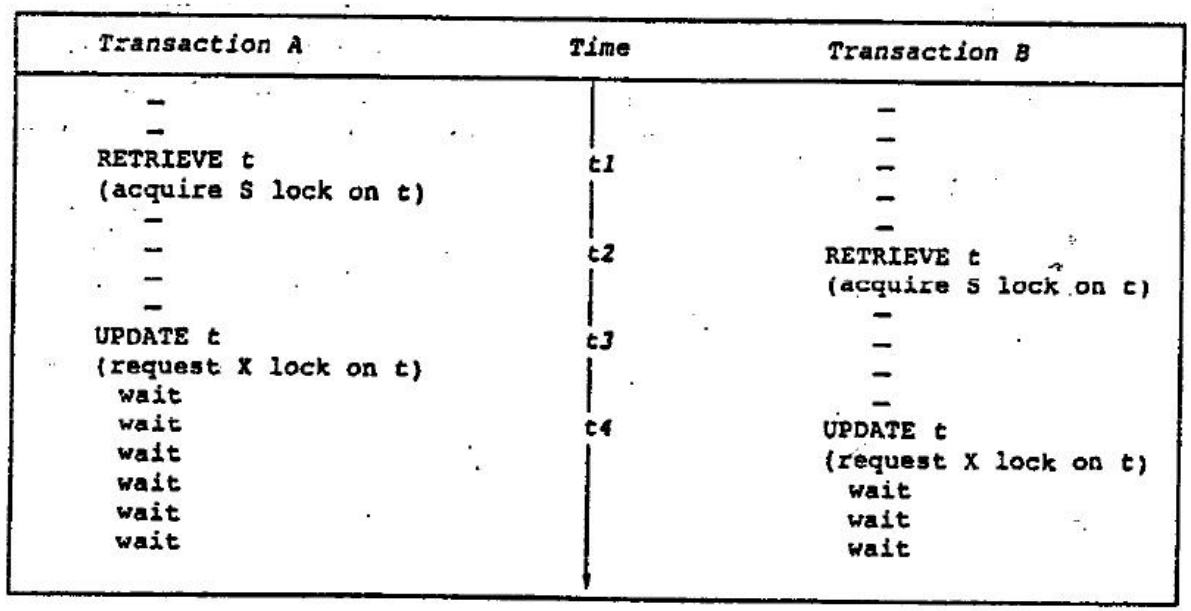
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Con bloqueos: Lectura repetible

Actualización perdida



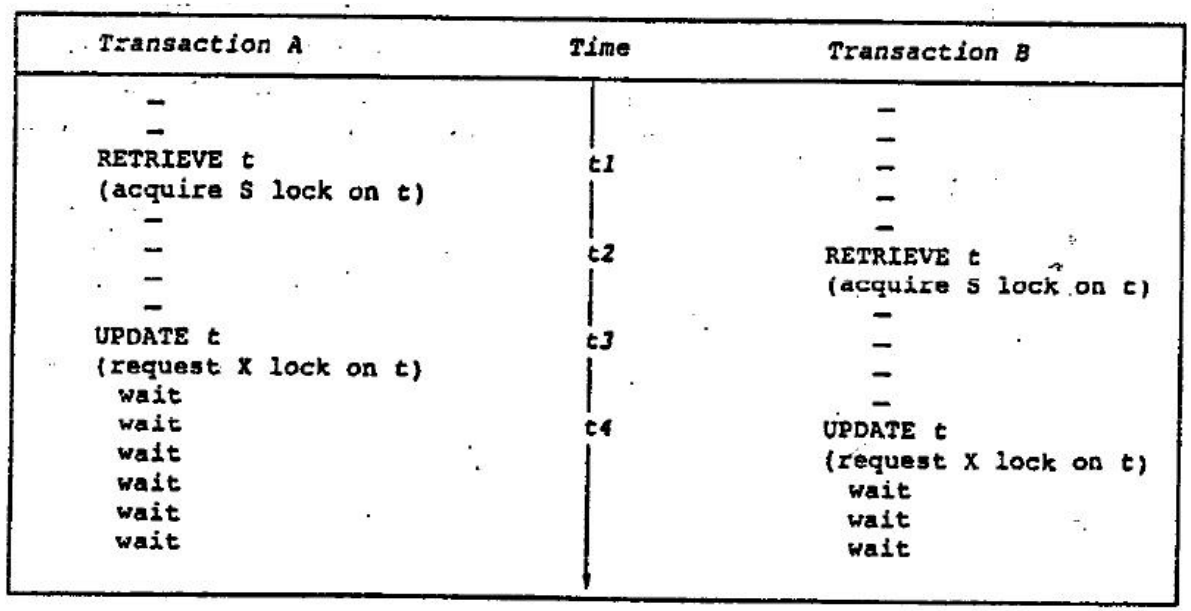
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–
wait	t4	UPDATE t
wait		(request X lock on t)
wait		wait
wait		wait
wait		wait

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Con bloqueos: Lectura repetible

Actualización perdida



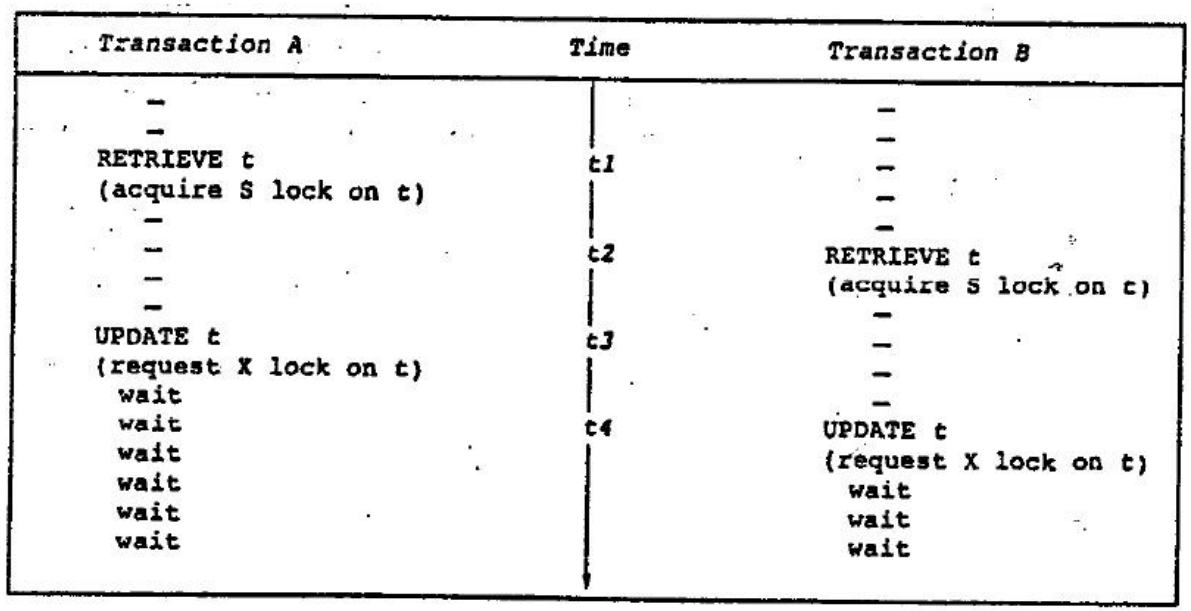
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–
wait	t4	UPDATE t
wait		(request X lock on t)
wait		wait
wait		wait
wait		wait

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Ejecutemos paso a paso

Con bloqueos: Lectura repetible

Actualización perdida



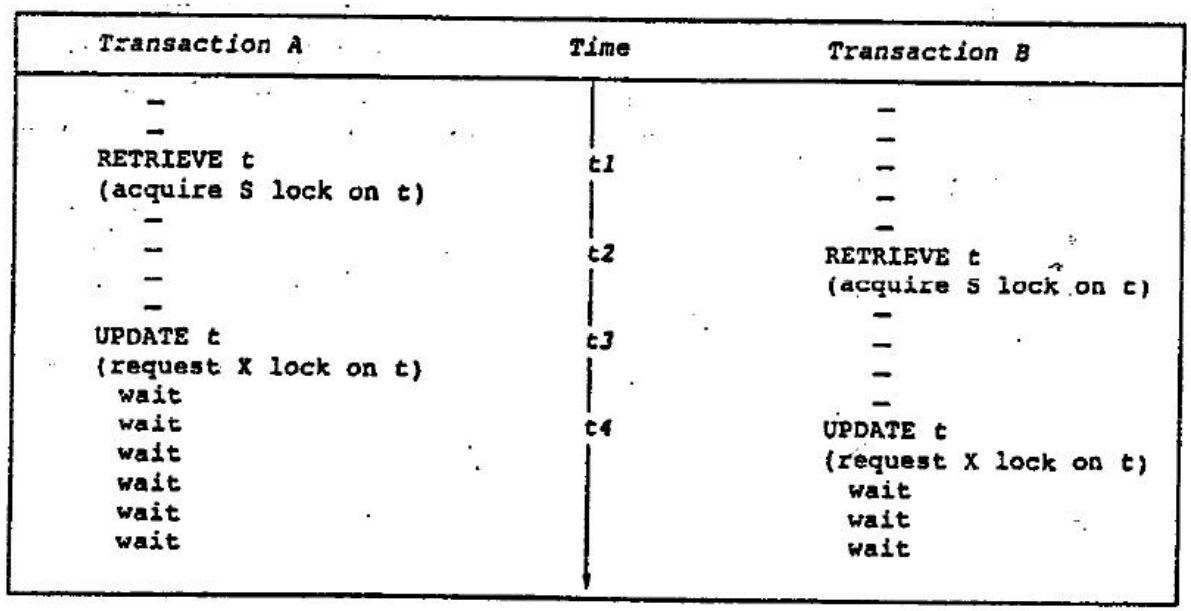
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–
wait	t4	UPDATE t
wait		(request X lock on t)
wait		wait
wait		wait
wait		wait

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Es correcto?

Con bloqueos: Lectura repetible

Actualización perdida



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Es correcto? Bloqueo mortal

Bloqueo mortal

¿Cómo se soluciona?

¿Cómo se detecta?

Bloqueo mortal

¿Cómo se soluciona?

Seleccionar una de las transacciones del bloqueo, **la víctima** y deshacerla, liberando sus bloqueos,

¿Cómo se detecta?

1. Mantener grafo de espera

   ¿Qué información guardamos?

   ¿Qué buscamos?

2. Control de tiempo

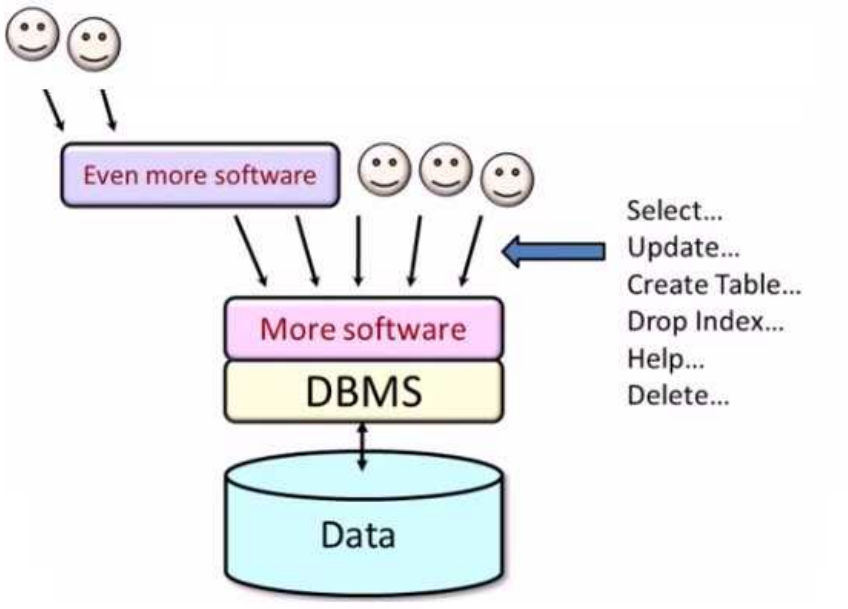
   ¿Qué información guardamos?

   ¿Qué buscamos?

Agenda

Agenda

Problemas-Solución



"MOOC Databases. Standford."

:(Dos aplicaciones que están ejecutando simultáneamente usan los mismos datos

Puede quedar la base de datos con datos incorrectos

Concurrencia

¿Cómo garantizamos una ejecución concurrente **correcta**?

:(El sistema falla cuando se está ejecutando

Puede quedar la base de datos en un estado desconocido (correcto o incorrecto)

Recuperación

¿Cómo volvemos a un estado **correcto** y conocido?

ADMINISTRACION DE TRANSACCIONES

Problemas

Fallas locales

Fallas globales

Problemas

Fallas locales

- Falla de operación

Fallas globales

- Falla del sistema
- Falla del medio

Agenda

Solución

Redundancia

Garantizar que cualquier parte de la información que contiene el sistema puede ser recuperada a partir de otra información guardada redundantemente en otro lugar del sistema

- Bitacora

  Mantiene registro de los cambios que cada transacción ha hecho a la base de datos

- Copias de respaldo

  Archivar la base de datos a un segundo disco

Solución

Punto de sincronización

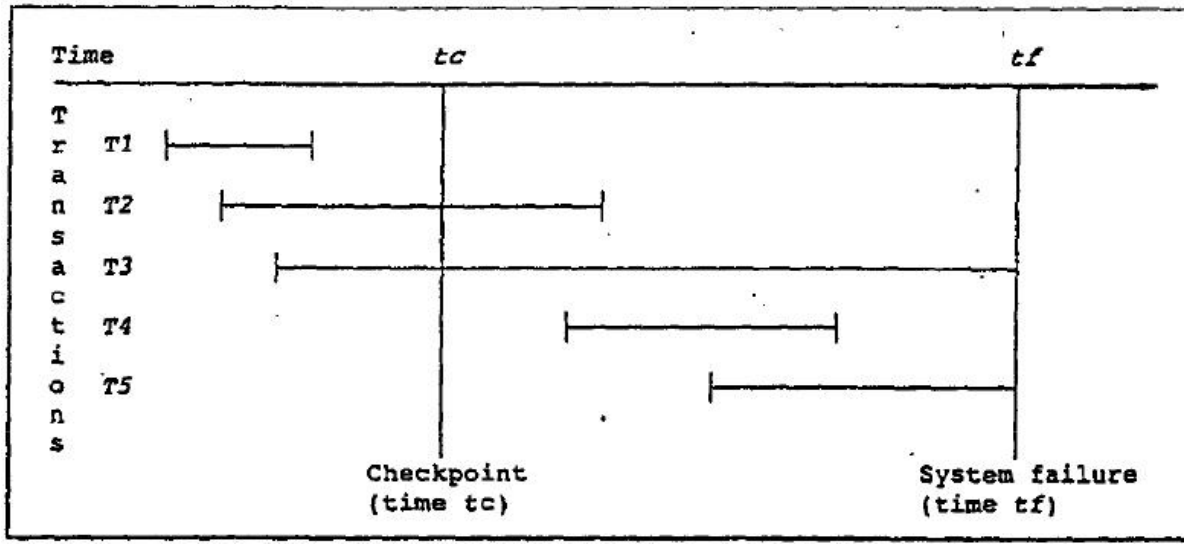
- Las instrucciones COMMIT y ROLLBACK establecen puntos de sincronización; son puntos en que la base de datos debería estar en un estado de consistencia

Puntos de verificación

- Escritura a disco, a determinados intervalos de tiempos prescritos

Soluciíon

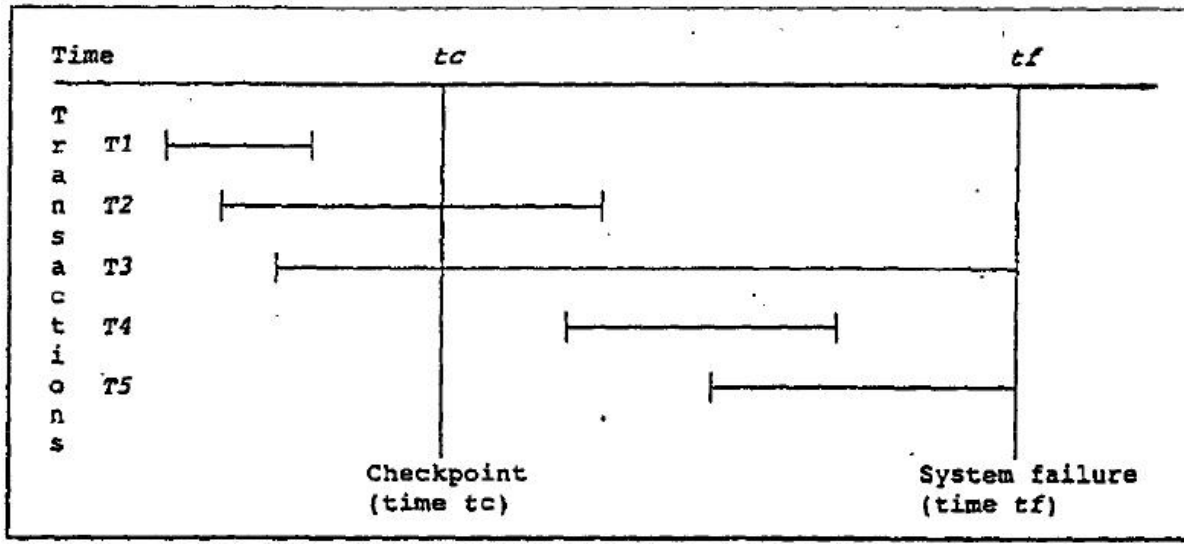
¿Cuáles son las acciones?



- ¿Qué queda listo?
- ¿Qué se debe rehacer?
- ¿ Qué se debe deshacer?

Soluciíon

¿Cuáles son las acciones?



- ¿Qué queda listo?

  Todo el trabajo realizado por las transacciones terminaron antes del punto de verificación

- ¿Qué se debe rehacer?

  Todo el trabajo realizado por las transacciones que terminaron satisfactoriamente antes de la caída

  Recuperación hacia adelante

- ¿ Qué se debe deshacer?

  Todo el trabajo realizado por las transacciones que no terminaron antes de la caída

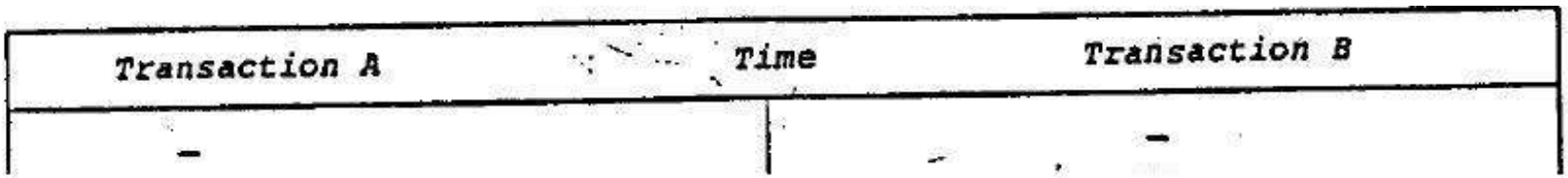
  Recuperación hacia atrás

Agenda

Problemas

Example

Actualización perdida

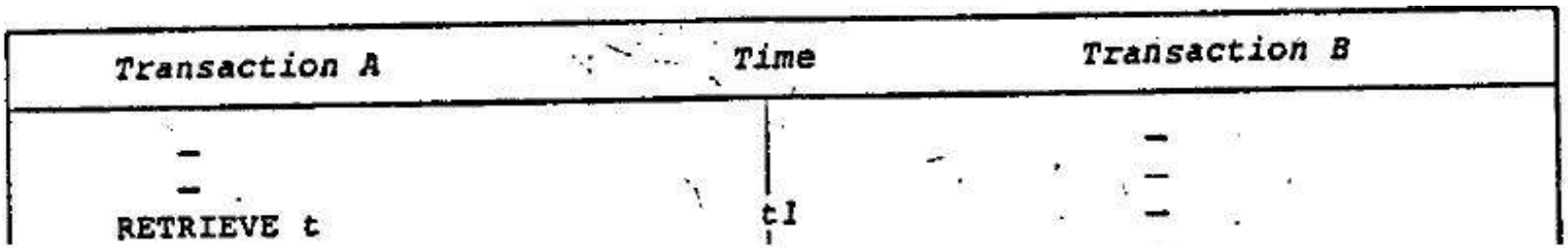


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Problemas

Example

Actualización perdida

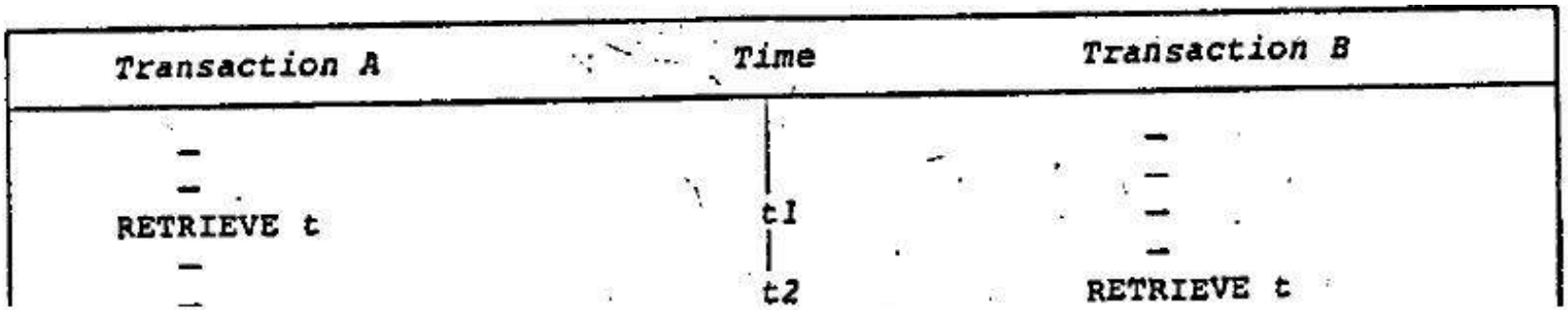


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Problemas

Example

Actualización perdida

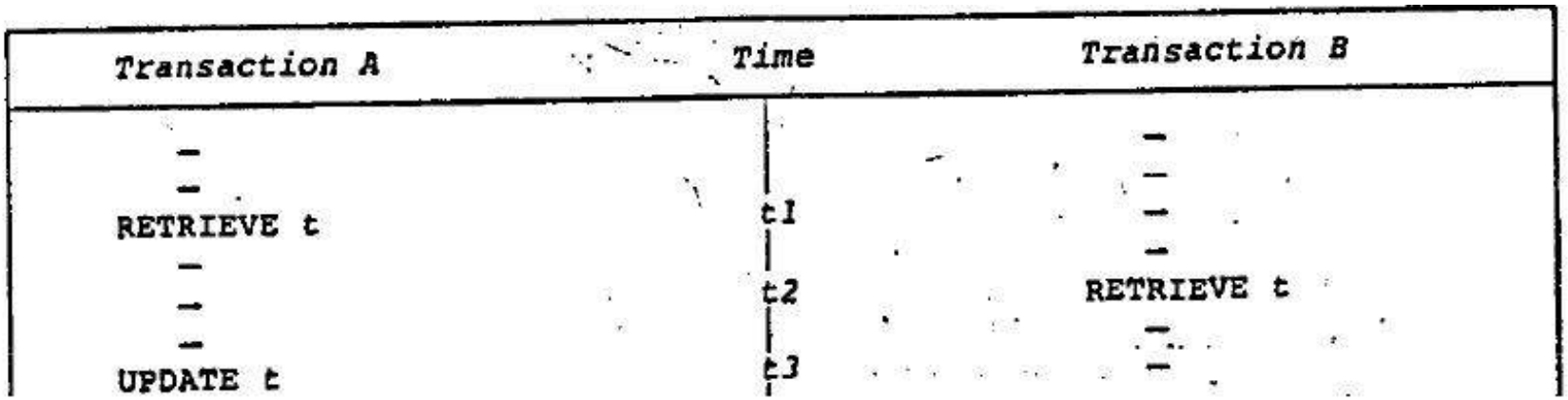


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Problemas

Example

Actualización perdida

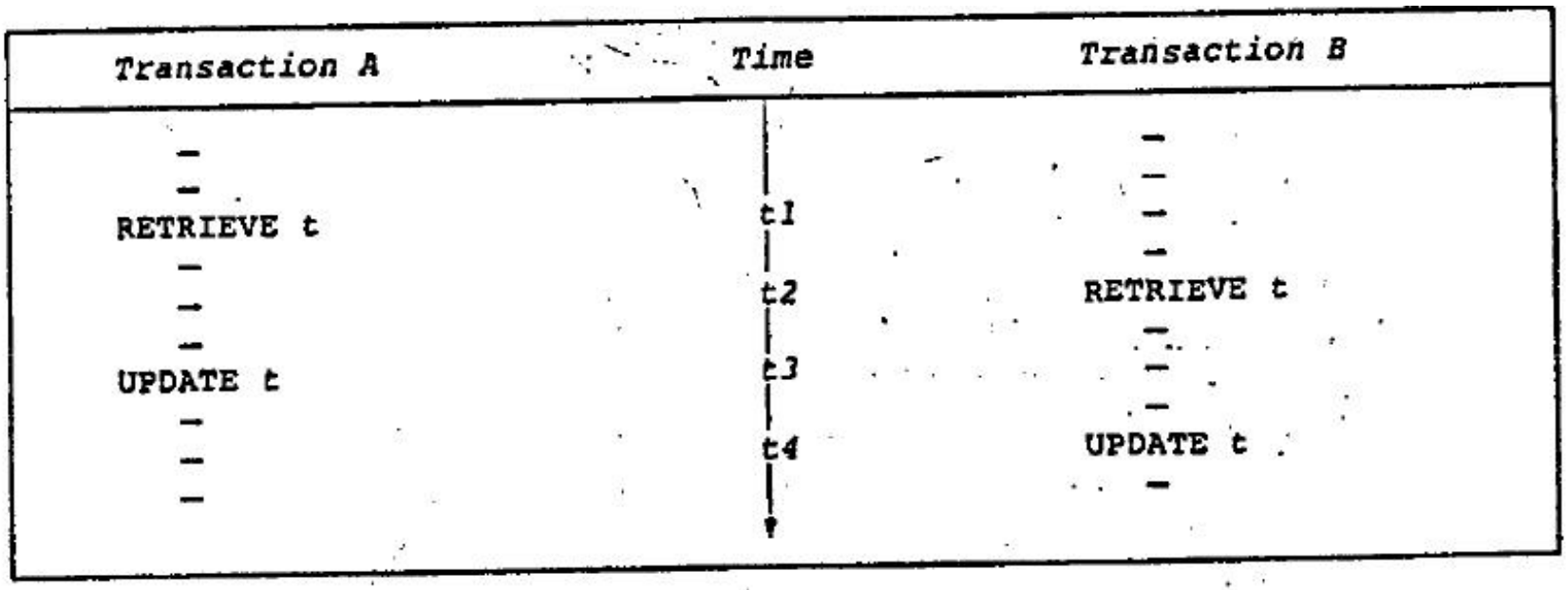


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Problemas

Example

Actualización perdida

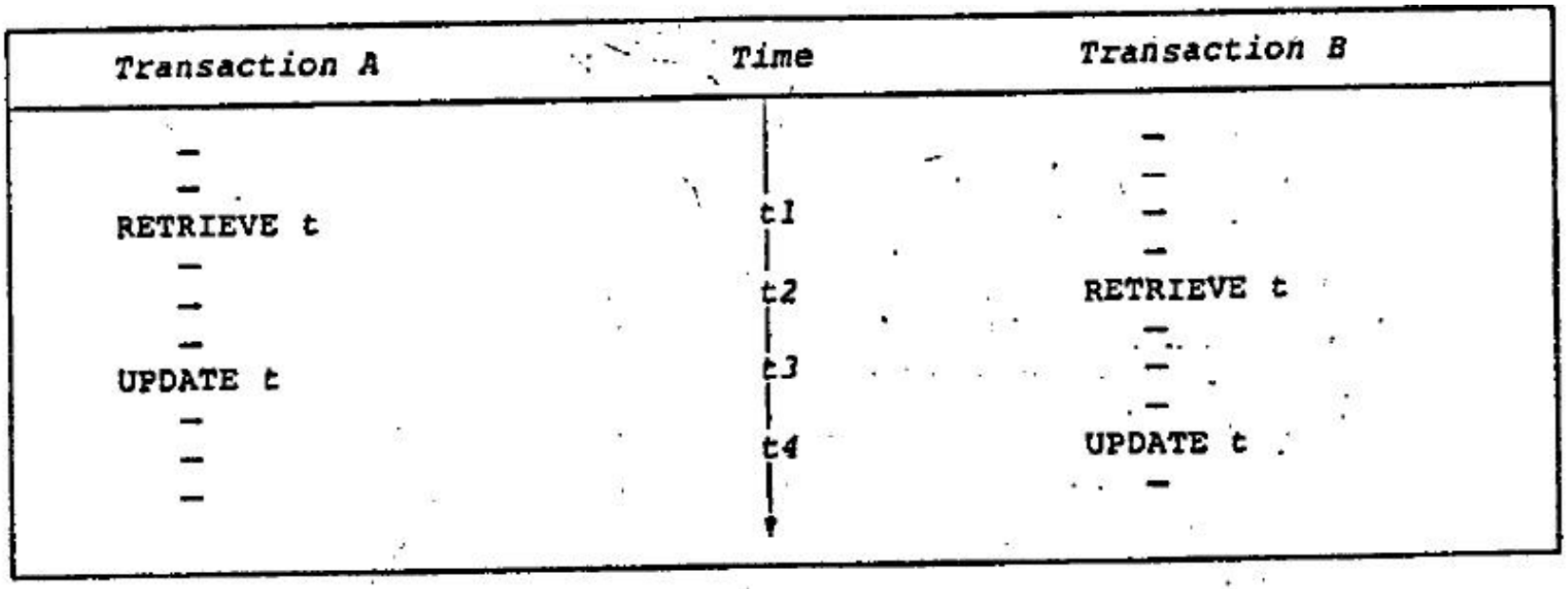


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Problemas

Example

Actualización perdida



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¡No es correcto! Lectura no repetible

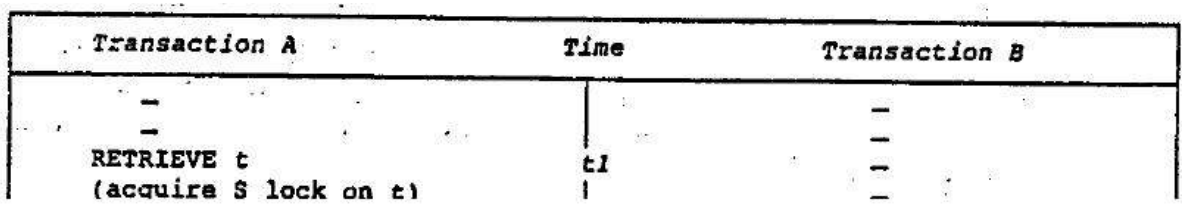
¿Solución?

Actualización perdida

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Solución?

Actualización perdida

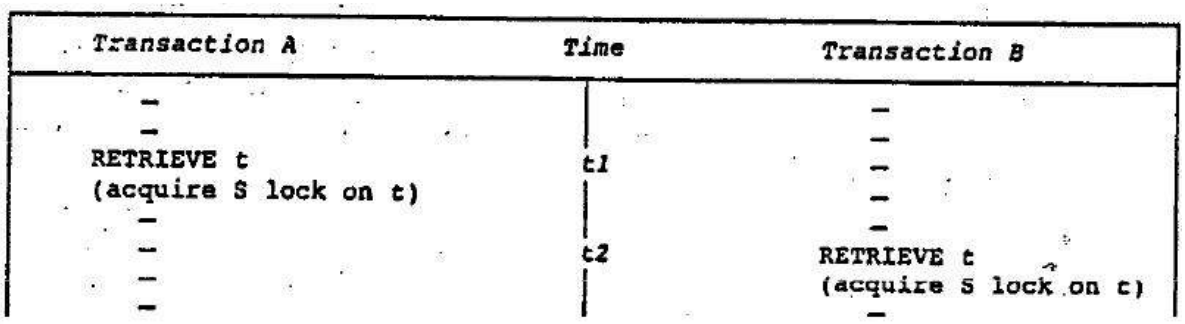


Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Solución?

Actualización perdida

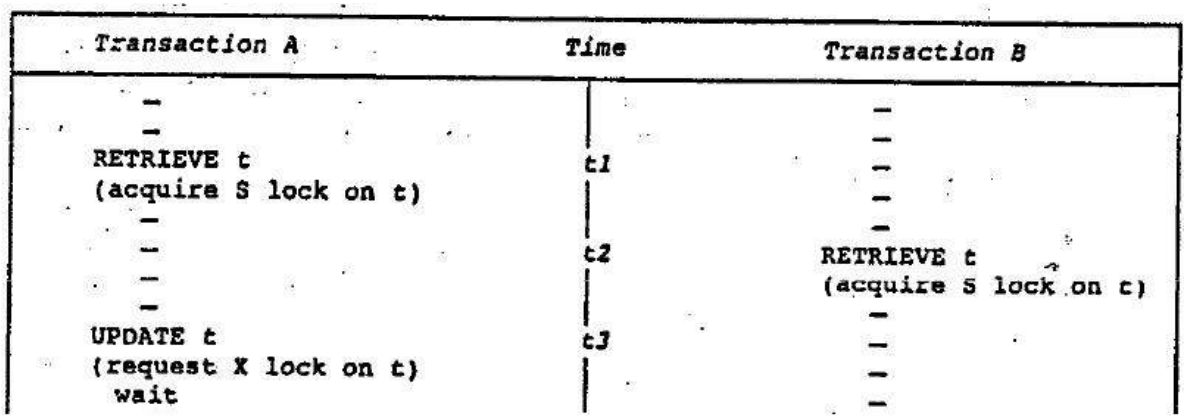


Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Solución?

Actualización perdida

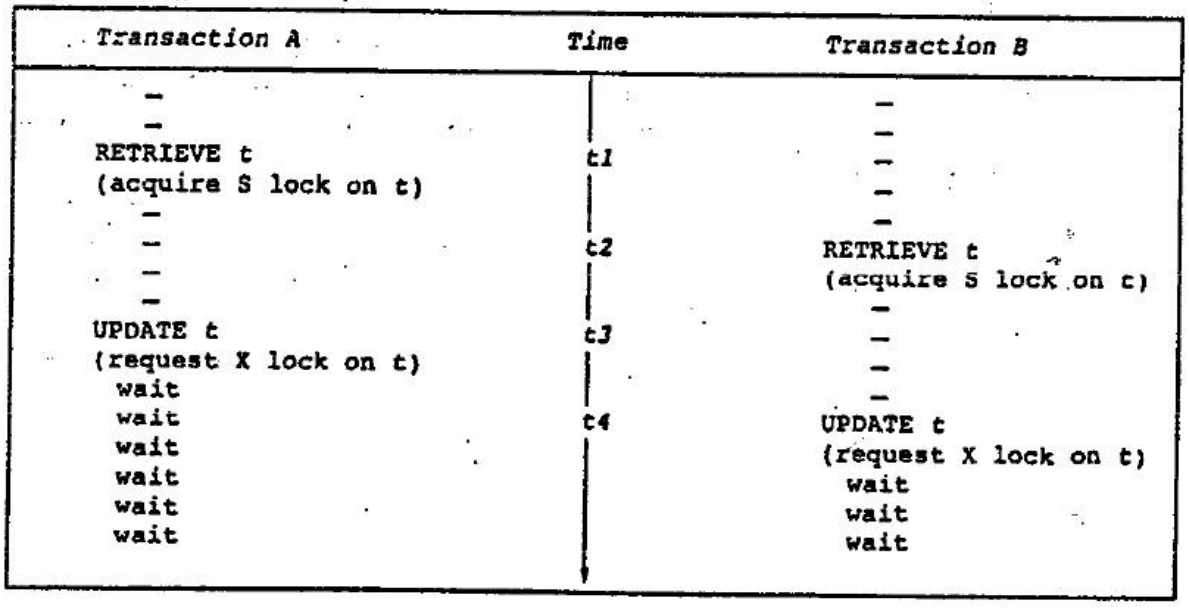


Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Solución?

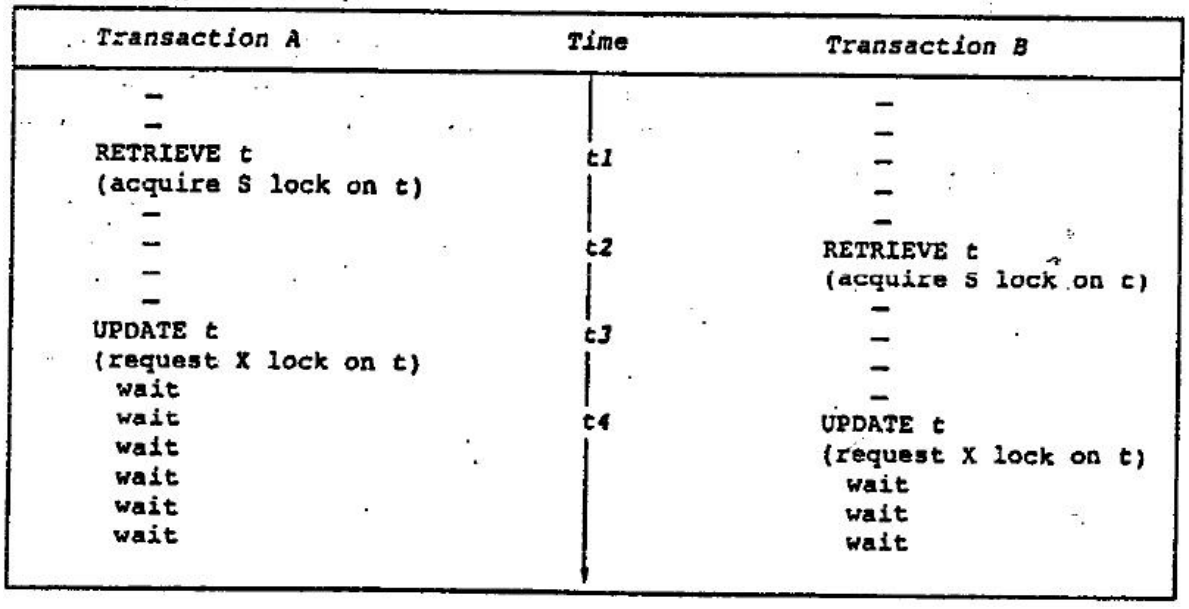
Actualización perdida



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

¿Solución?

Actualización perdida



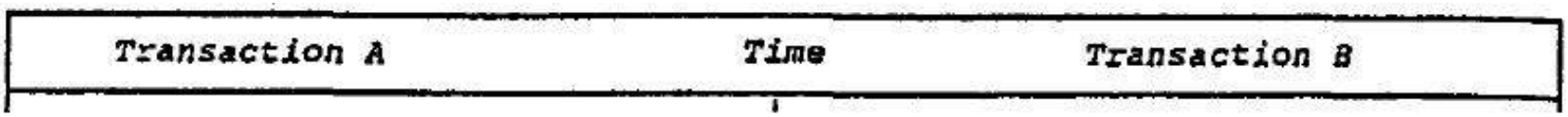
Transaction A	Time	Transaction B
–		–
–		–
RETRIEVE t	t1	–
(acquire S lock on t)		–
–		–
–	t2	RETRIEVE t
–		(acquire S lock on t)
–		–
UPDATE t	t3	–
(request X lock on t)		–
wait		–
wait	t4	UPDATE t
wait		(request X lock on t)
wait		wait
wait		wait
wait		wait

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consignación de 250.000
- La transacción B es un retiro de 500.000

Bloqueo mortal

Problemas

Dependencia no confirmada



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta el saldo
- La transacción B trata de hacer un retiro de 500.000 pero se arrepiente

Problemas

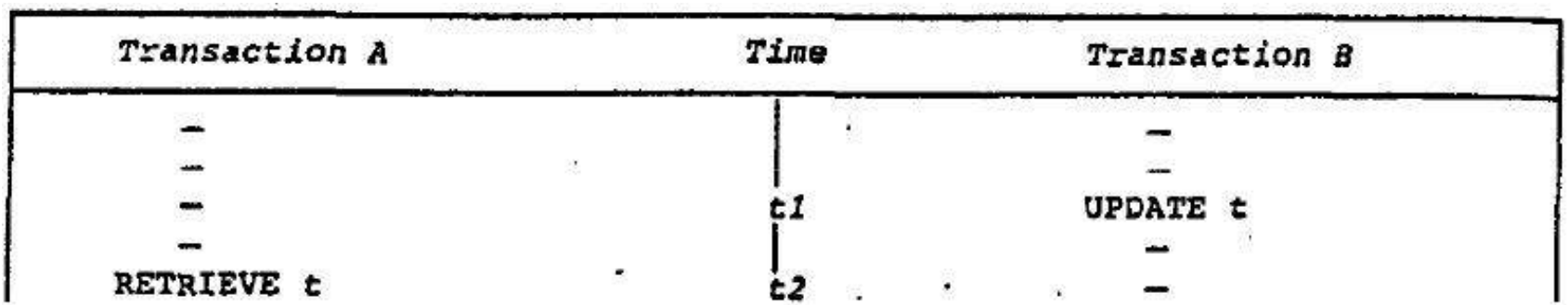
Dependencia no confirmada

Transaction A	Time	Transaction B
–		–
–		–
–	t1	UPDATE t

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta el saldo
- La transacción B trata de hacer un retiro de 500.000 pero se arrepiente

Problemas

Dependencia no confirmada

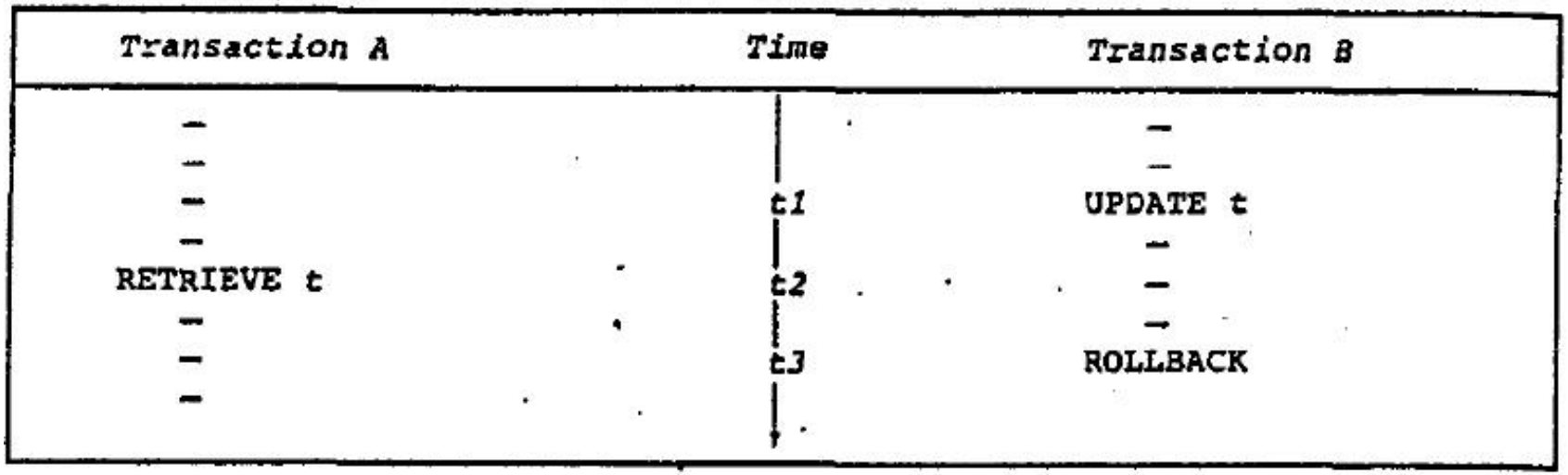


Transaction A	Time	Transaction B
–		–
–		–
–	t1	UPDATE t
–		–
RETRIEVE t	t2	–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta el saldo
- La transacción B trata de hacer un retiro de 500.000 pero se arrepiente

Problemas

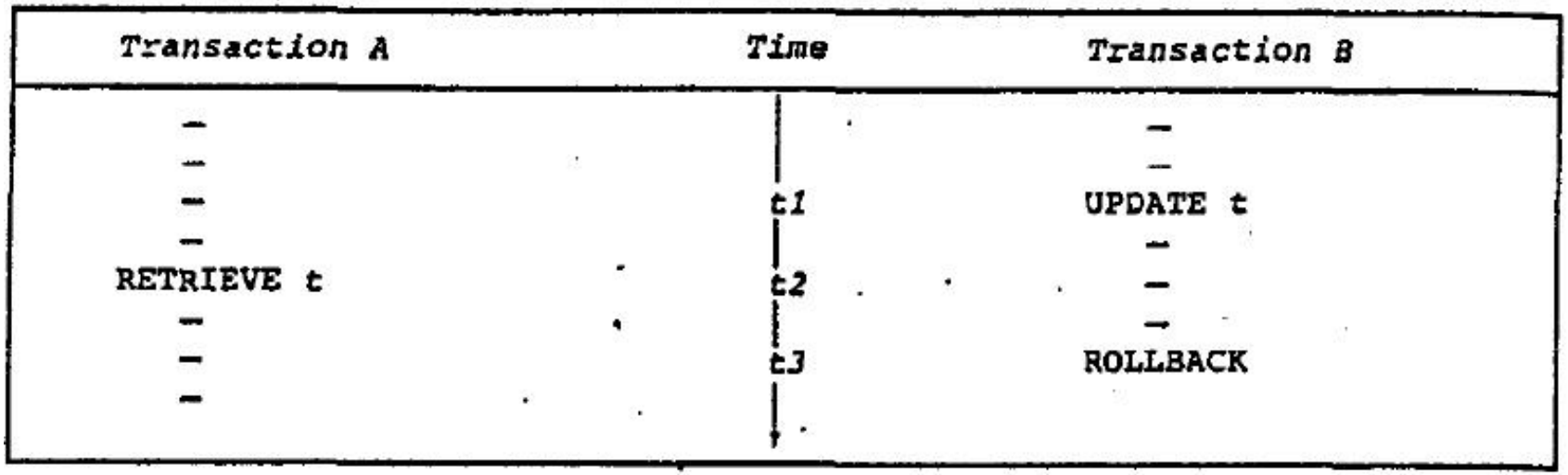
Dependencia no confirmada



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta el saldo
- La transacción B trata de hacer un retiro de 500.000 pero se arrepiente

Problemas

Dependencia no confirmada

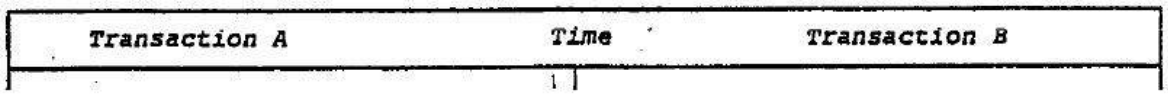


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta el saldo
- La transacción B trata de hacer un retiro de 500.000 pero se arrepiente

¡ No es correcto!. Lectura sucia

¿Solución?

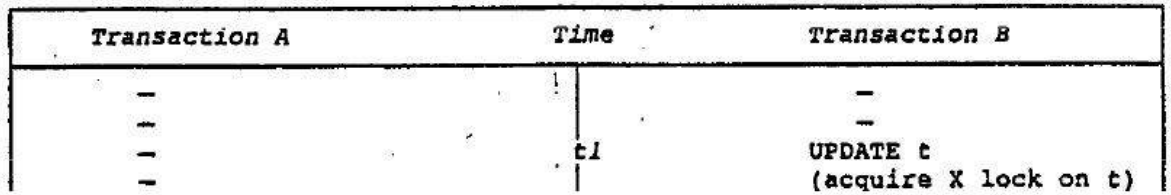
Dependencia no confirmada



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¿Solución?

Dependencia no confirmada

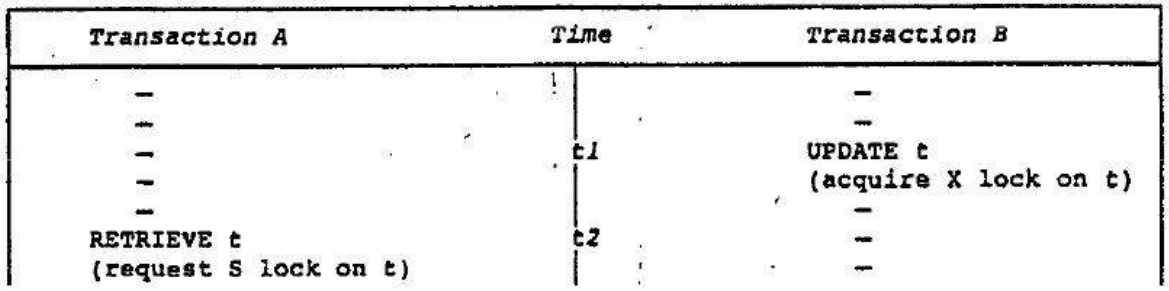


Transaction A	Time	Transaction B
–		–
–		–
–	t1	UPDATE t
–		(acquire X lock on t)

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¿Solución?

Dependencia no confirmada



Transaction A	Time	Transaction B
–		–
–		–
–	t1	UPDATE t
–		(acquire X lock on t)
–		–
RETRIEVE t	t2	–
(request S lock on t)		–

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¿Solución?

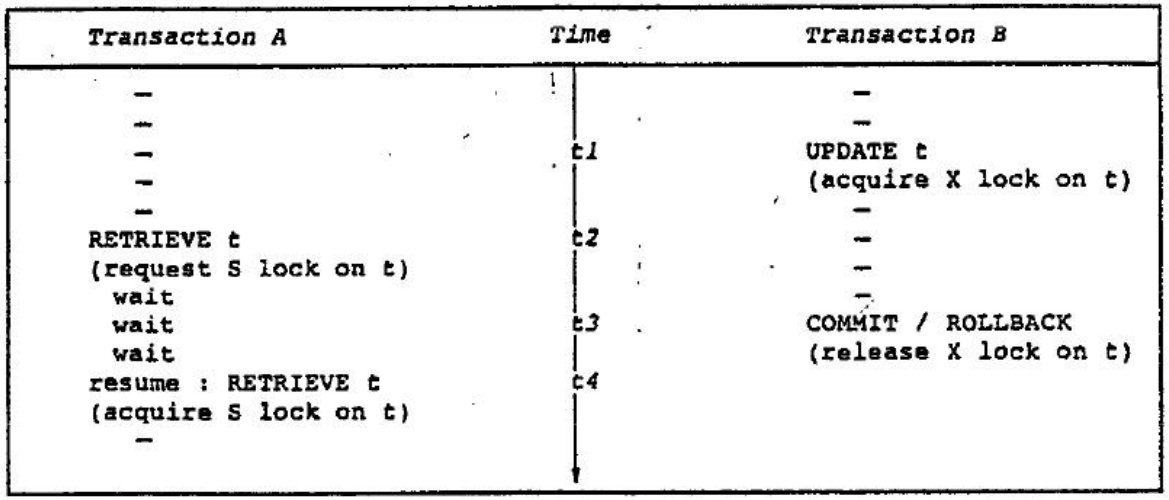
Dependencia no confirmada

Transaction A	Time	Transaction B
–		–
–		–
–	t1	UPDATE t
–		(acquire X lock on t)
–		–
RETRIEVE t	t2	–
(request S lock on t)		–
wait		–
wait	t3	COMMIT / ROLLBACK
wait		(release X lock on t)

- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¿Solución?

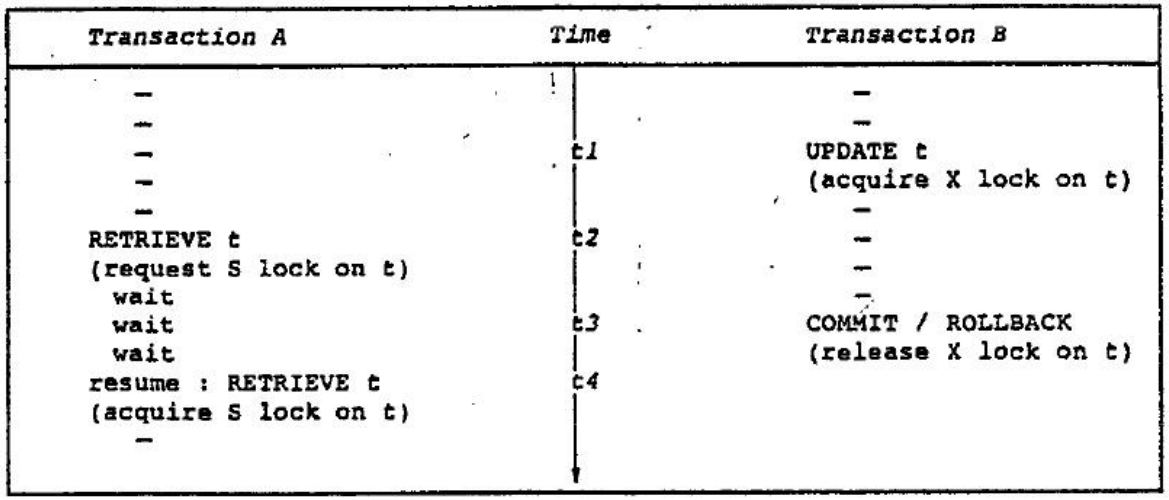
Dependencia no confirmada



- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¿Solución?

Dependencia no confirmada

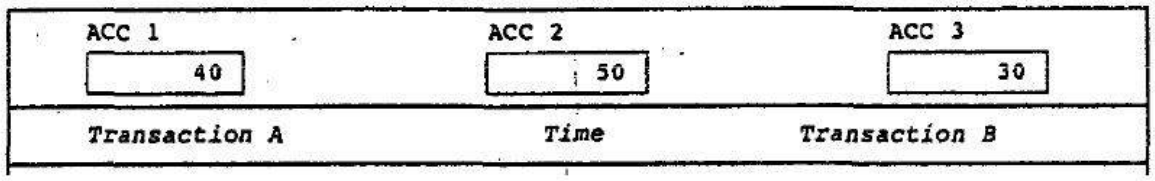


- Tenemos una cuenta **t** con saldo 1.000.000
- La transacción A es una consulta de saldo
- La transacción B es un retiro de 500.000

¡Es correcto!

Problemas

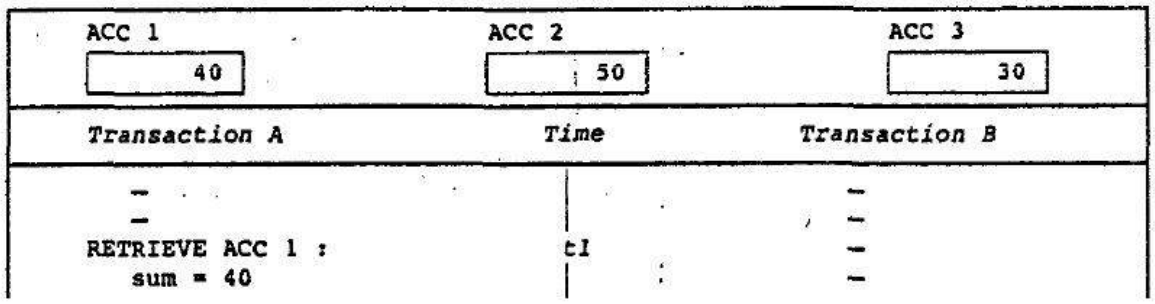
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

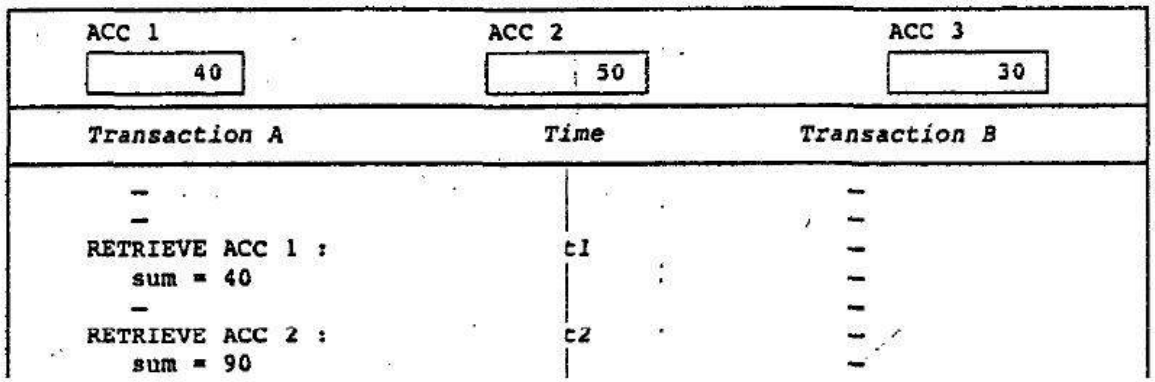
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

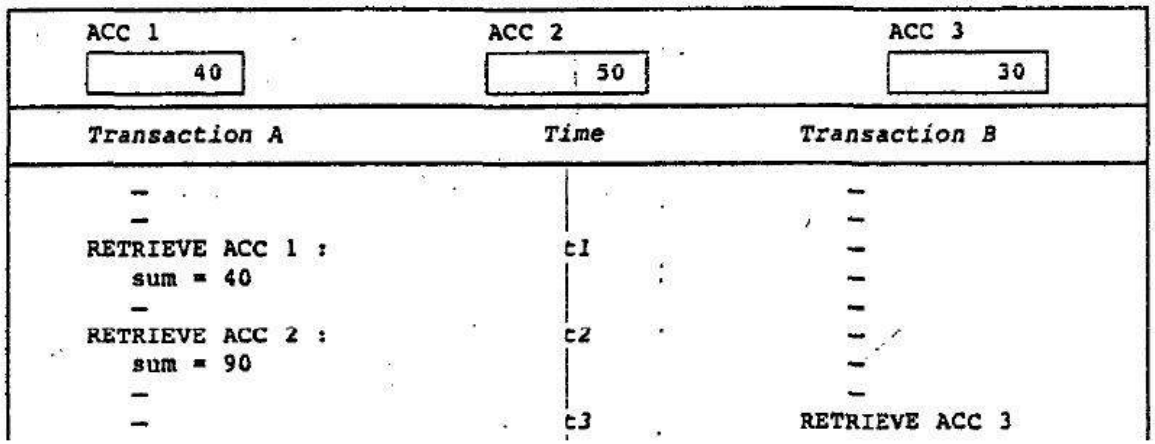
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

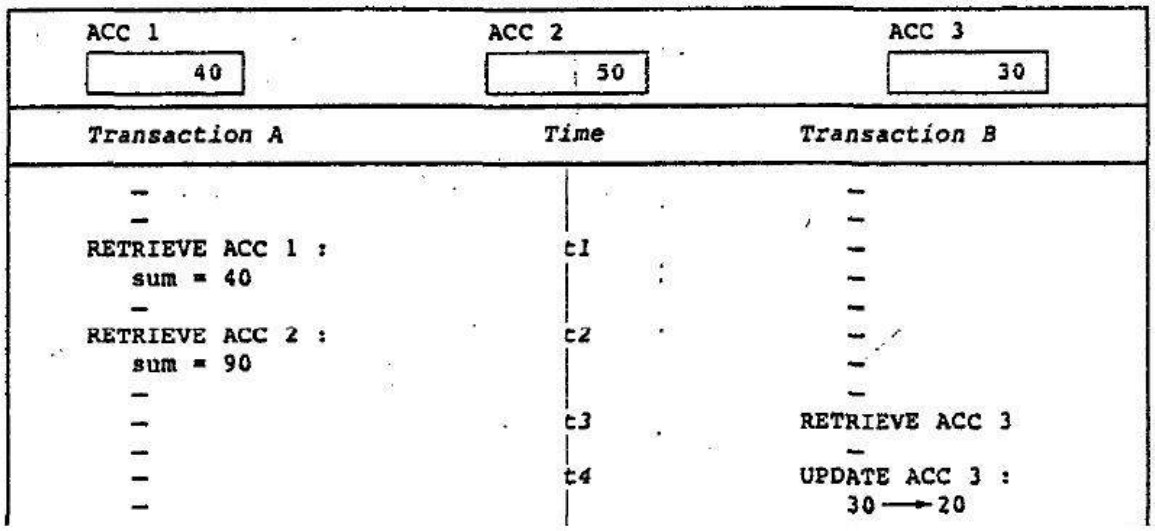
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

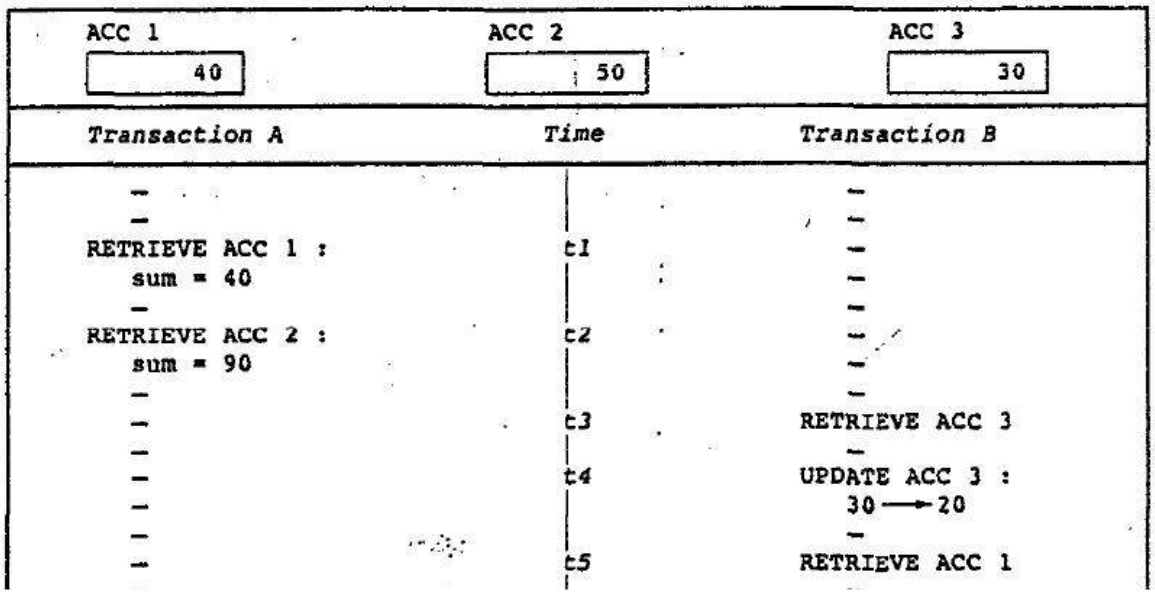
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

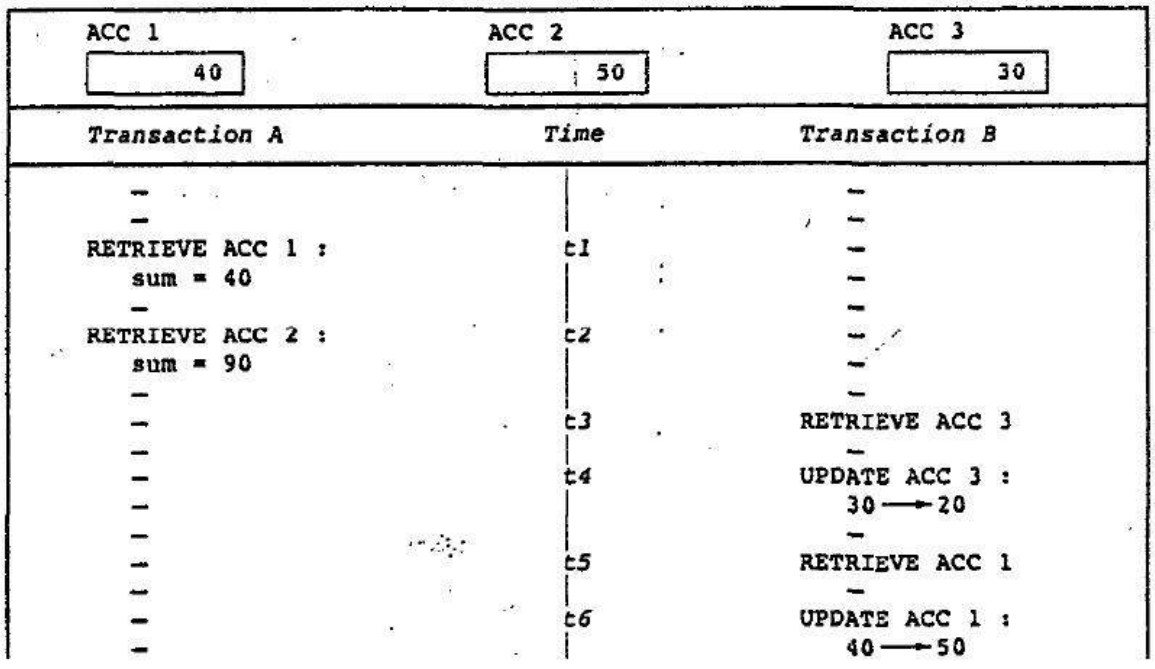
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

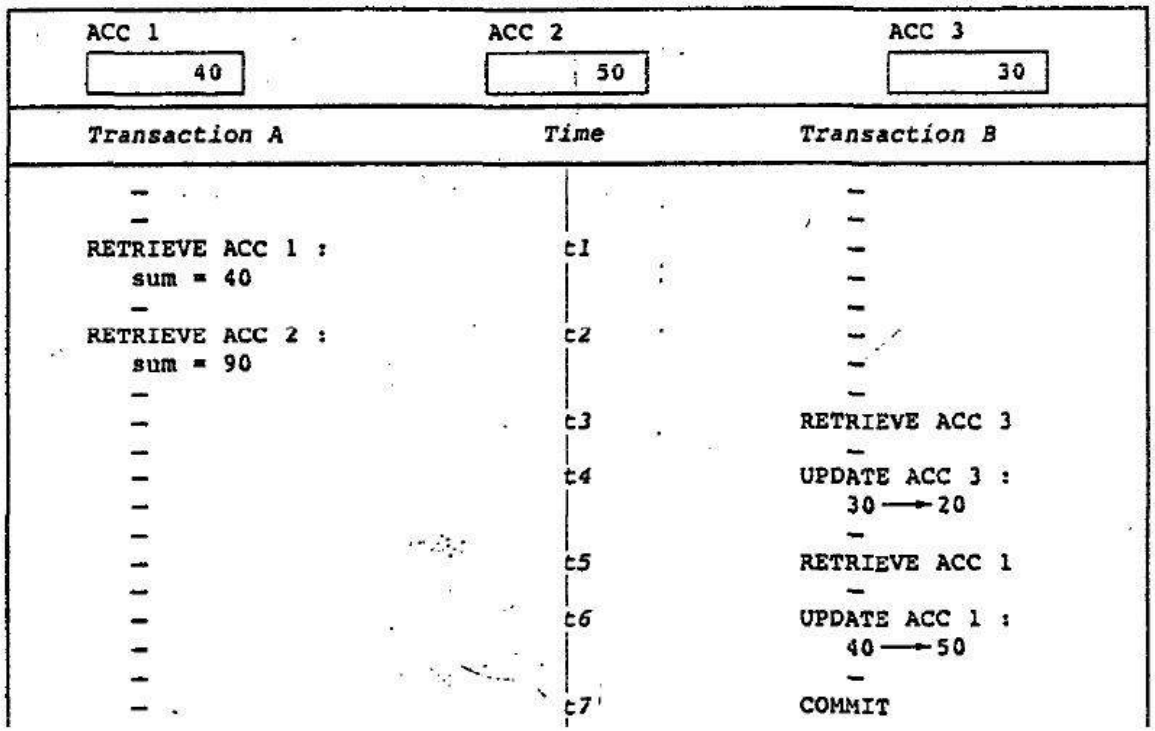
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

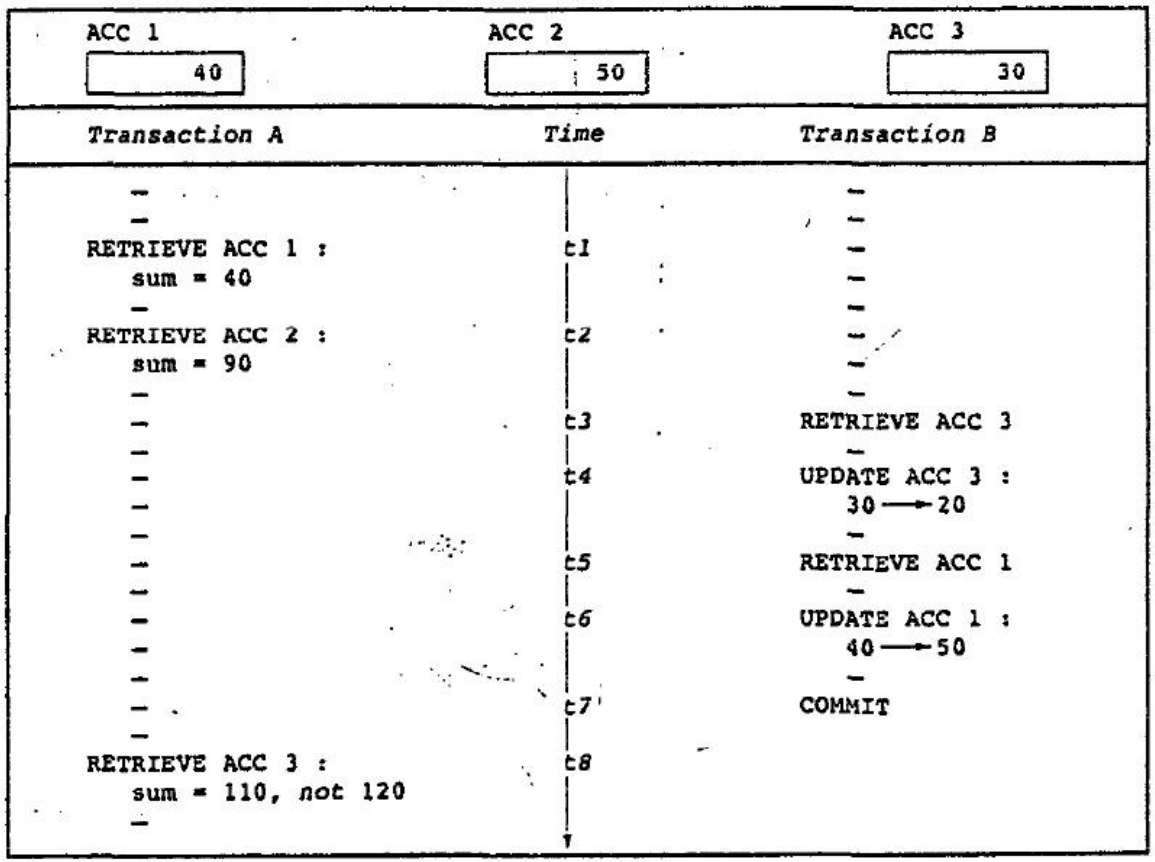
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

Problemas

Análisis inconsistente

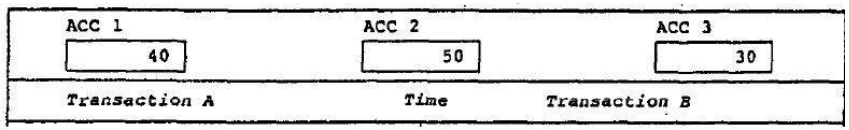


- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¡No es correcto! Lectura no repetible

¿Solución?

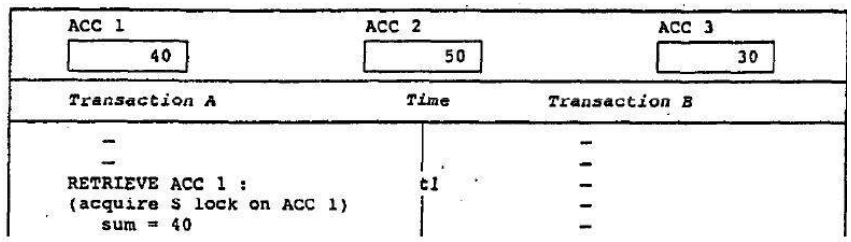
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

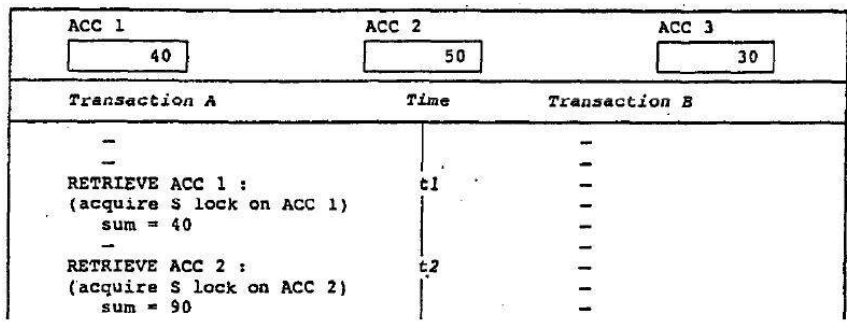
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

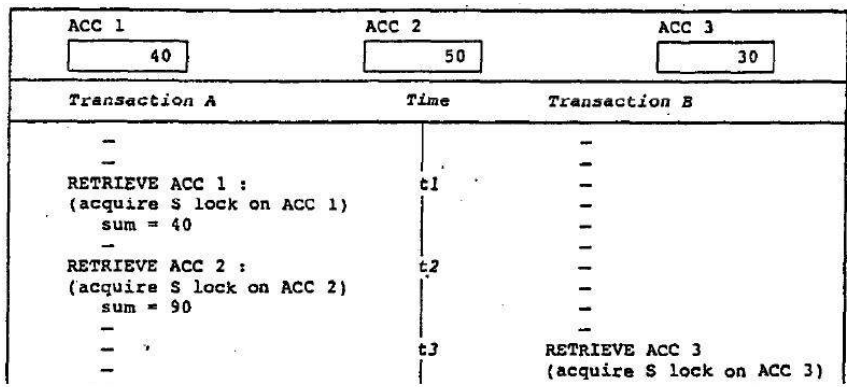
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

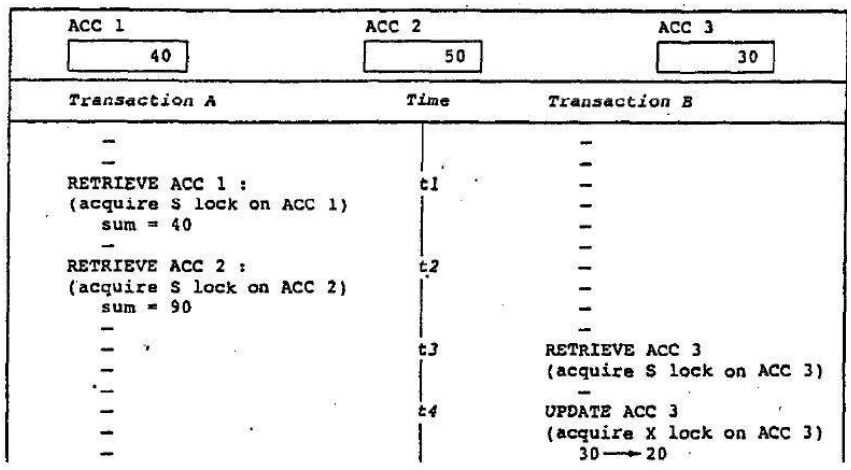
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

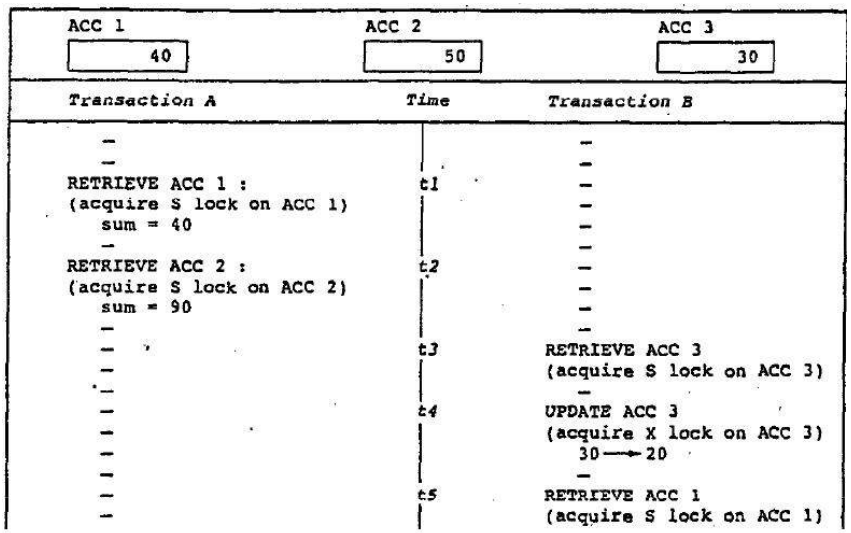
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

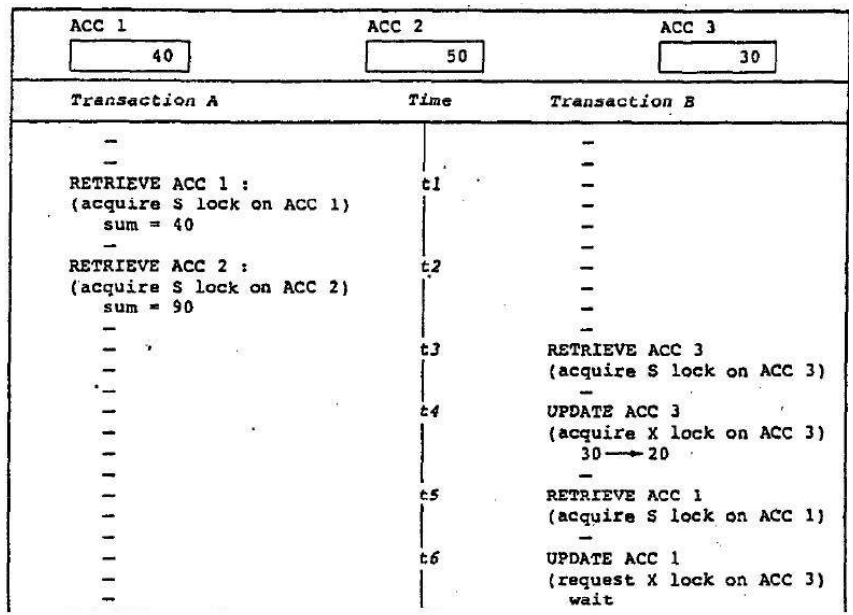
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

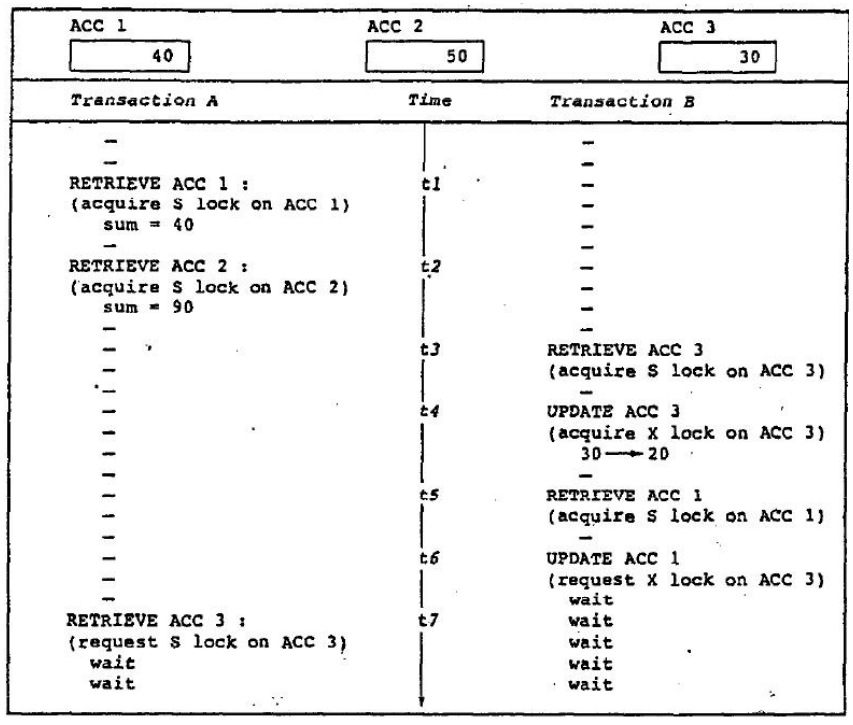
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

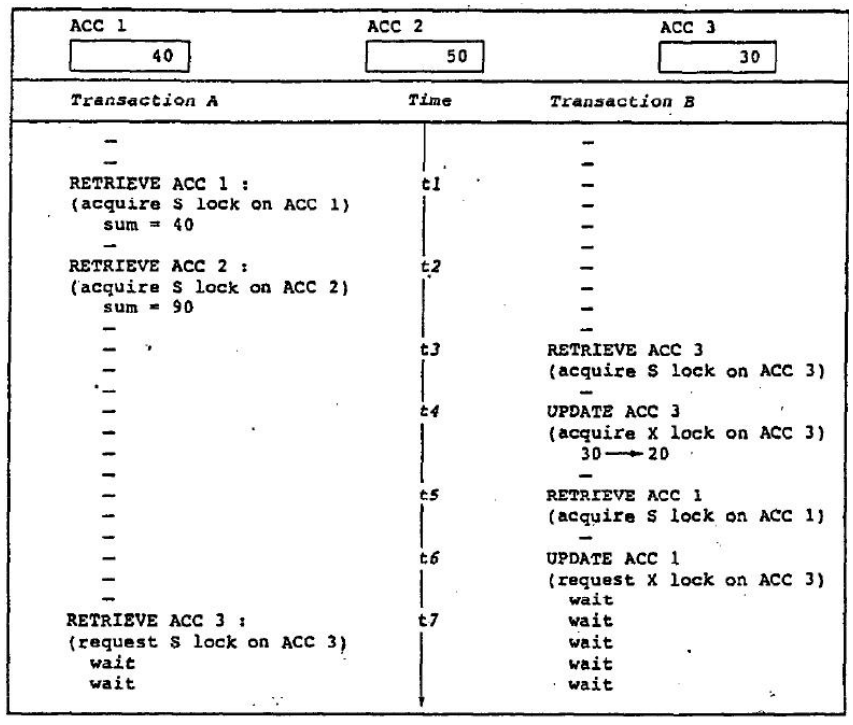
Análisis inconsistente



- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

¿Solución?

Análisis inconsistente

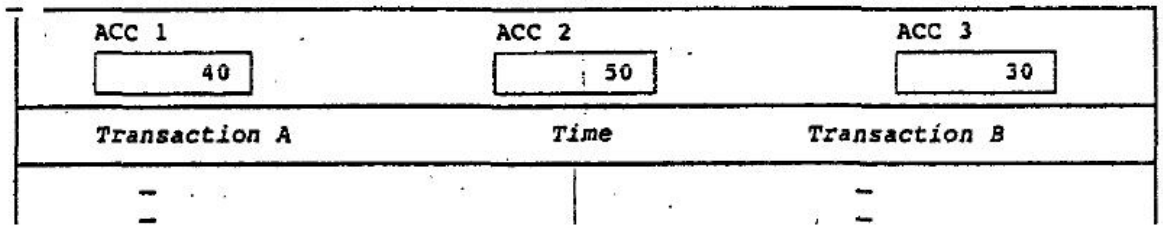


- La transacción A está calculando el saldo de las cuentas
- La transacción B está transfiriendo 10 de la cuenta ACC3 a la cuenta ACC1

  BLOQUEO MORTAL

Problemas

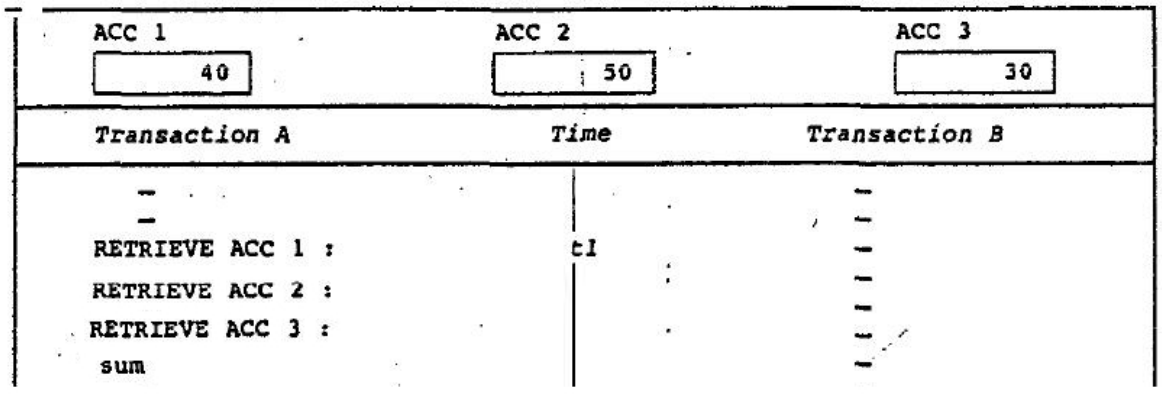
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

Problemas

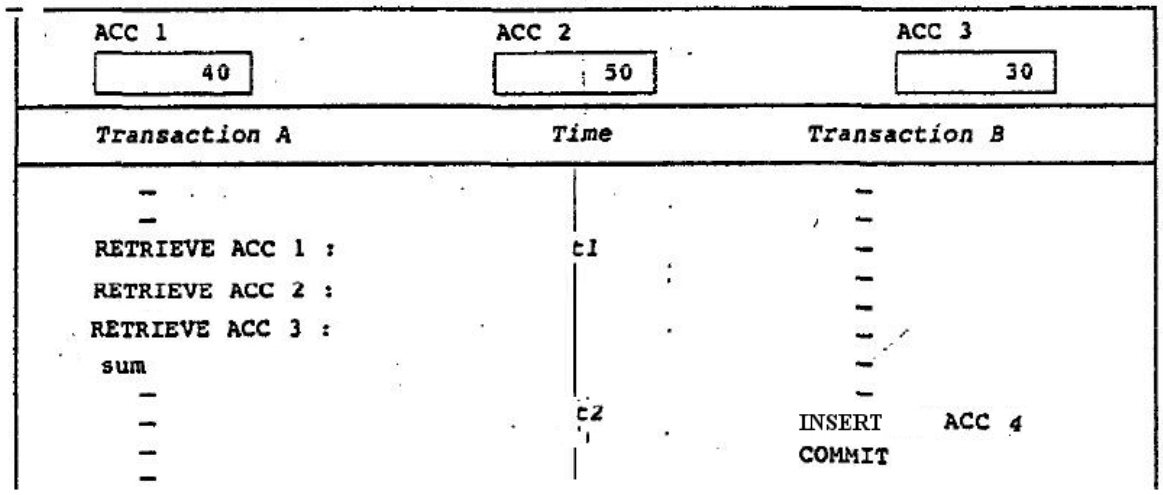
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

Problemas

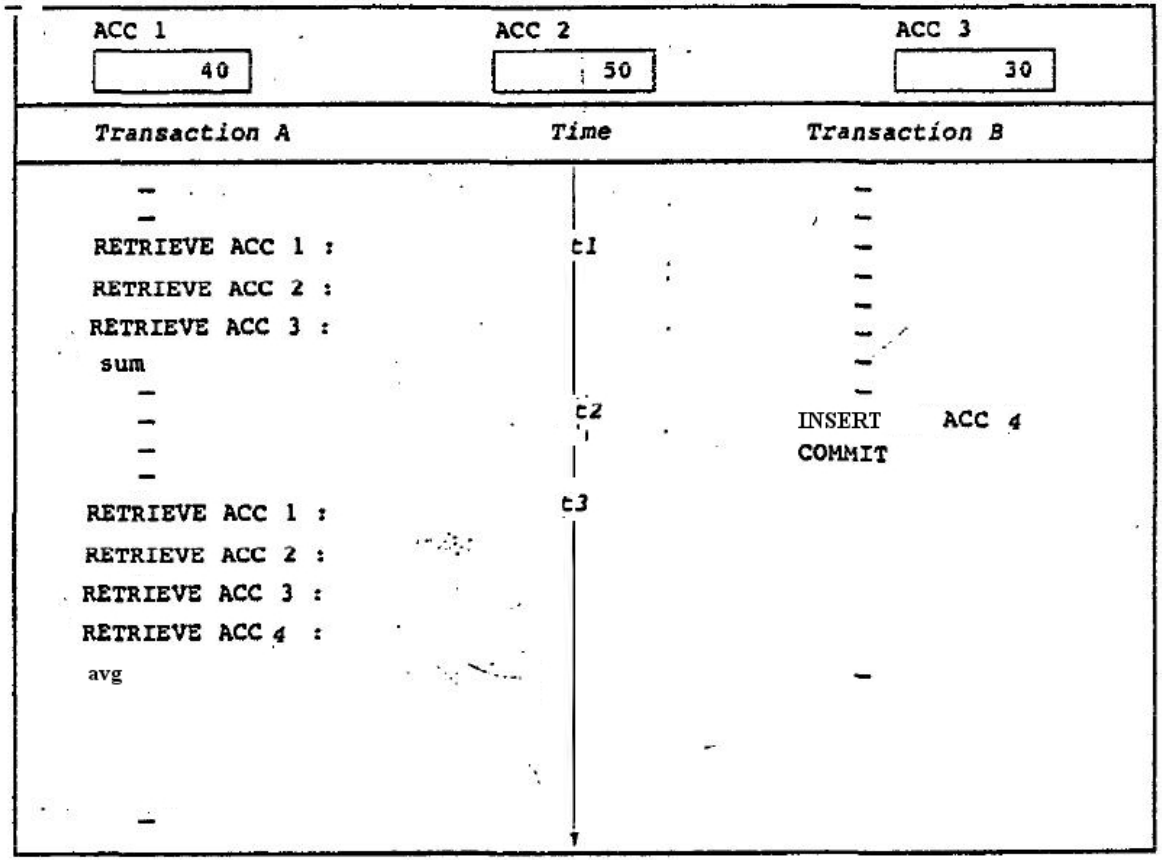
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

Problemas

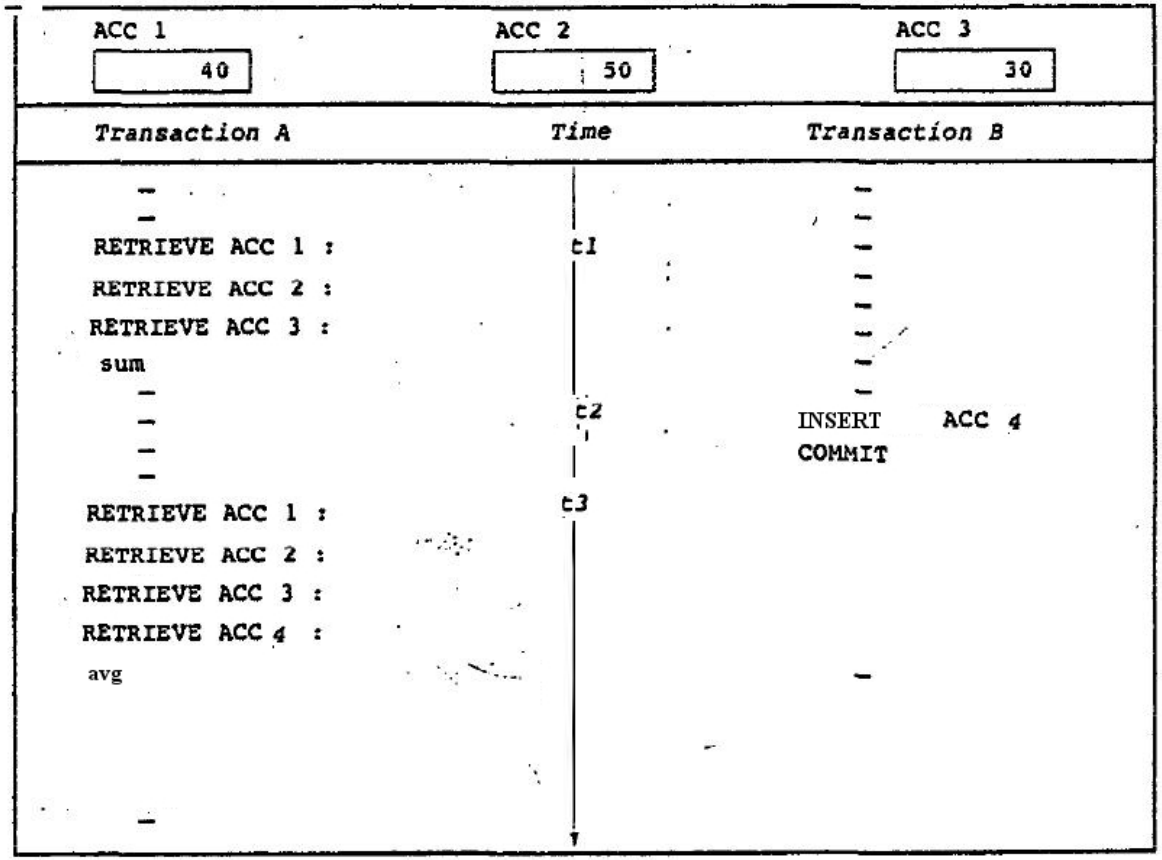
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

Problemas

¿?

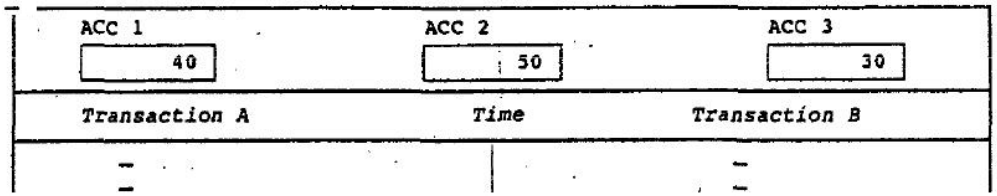


- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

  ¡No es correcto! Fantasmas.

¿Solución?

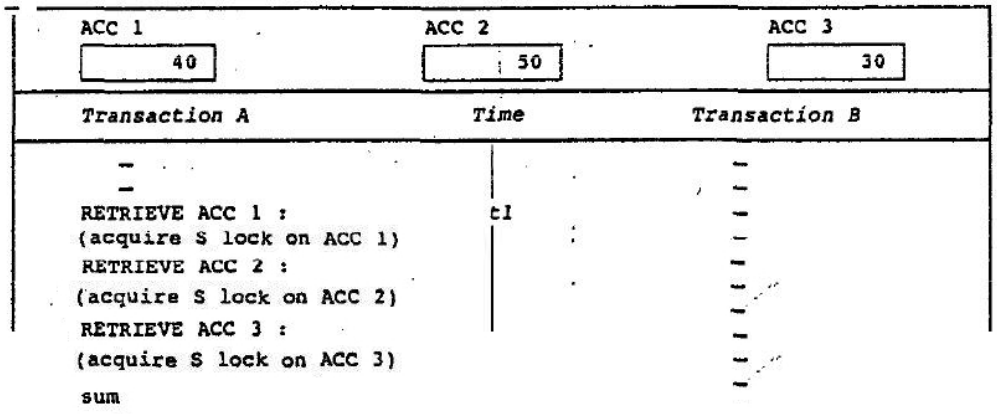
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

¿Solución?

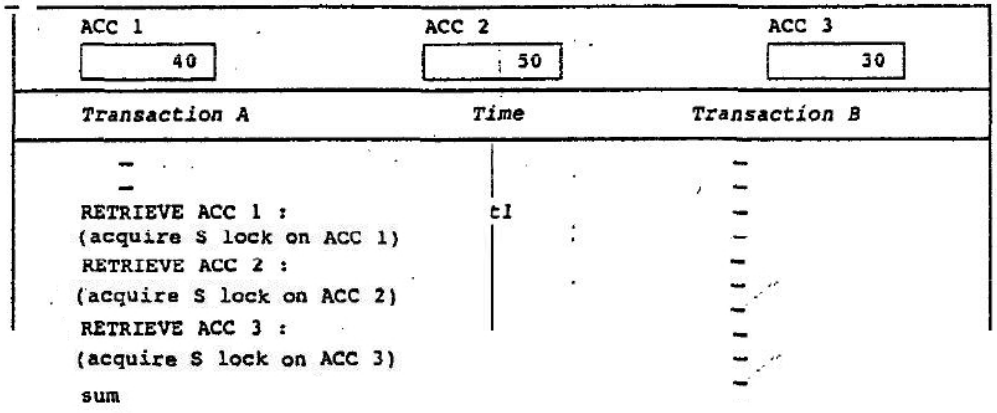
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

¿Solución?

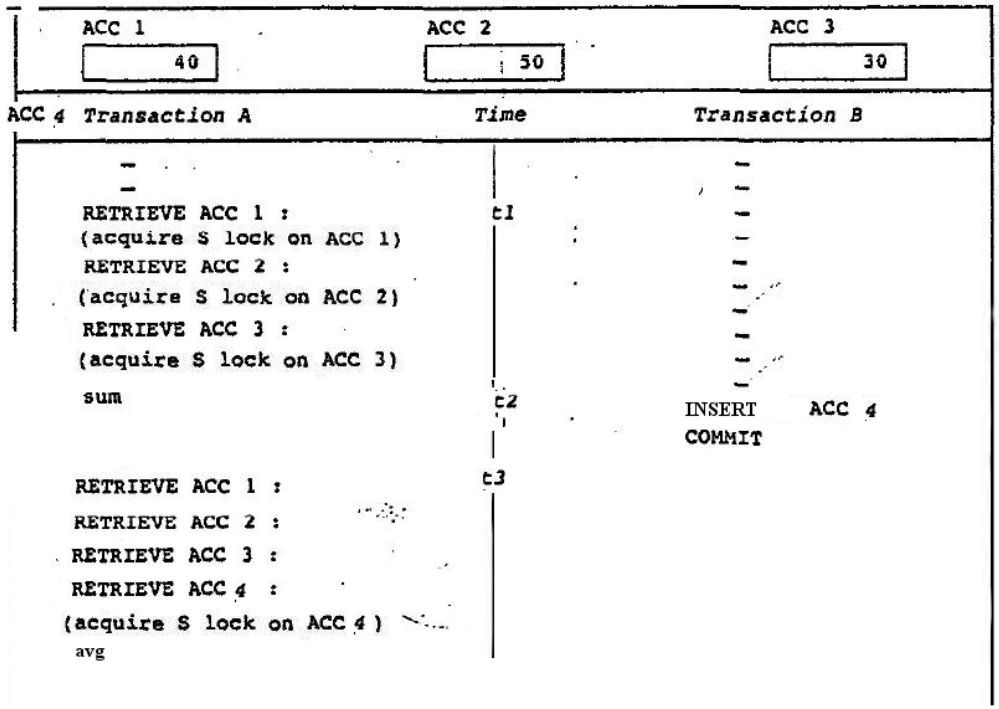
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

¿Solución?

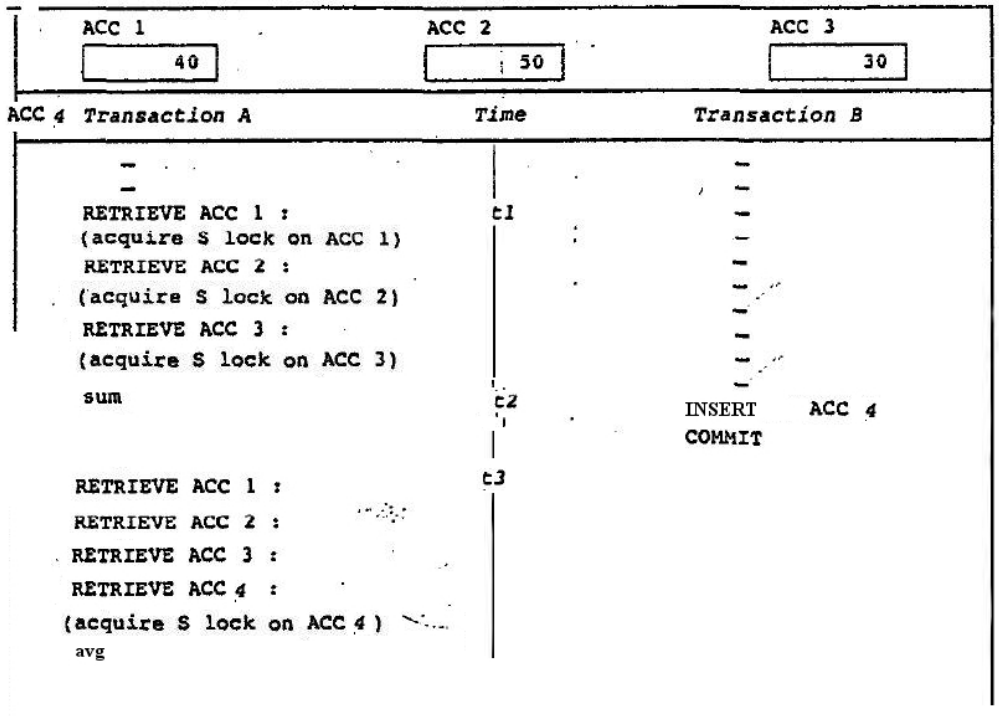
¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

¿Solución?

¿?



- La transacción A está calculando primero la suma y luego el promedio de las cuentas
- La transacción B está adicionando una nueva cuenta ACC4 100

¡No es correcto! Fantasmas.