

Bases de Datos

Unidad 5

Instructor: M.C. Luis Basto Díaz
UADY, FMAT
ITESM, Campus Monterrey
Email: luis.basto@uady.mx

Unidad 5

Control de Concurrencia

Instructor: M.C. Luis Basto Díaz
luis.basto@uady.mx

Control de Concurrencia

- Planes de ejecución
- Seriabilidad
- Bloqueo en dos fases

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Planes de ejecución

- Asegurar la consistencia a pesar de la concurrencia de las transacciones requiere trabajo extra.
- Es más sencillo que las transacciones se ejecuten secuencialmente.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Planes de ejecución

- Razones para permitir la concurrencia:
 - Aumento en la productividad (throughput), número de transacciones que se pueden ejecutar en un tiempo determinado
 - Mezcla de las transacciones, unas cortas y otras extensas (problema de esperar mientras termina una transacción para iniciar otra)

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Ejemplo: Transacción T1

- Sea T_1 una transacción para transferir \$1000 de la cuenta A a la cuenta B. Se define como sigue:

T1: Leer(A);
A:=A - 1000
Escribir(A);
Leer(B)
B:=B + 1000
Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Ejemplo: Transacción T2

- Sea T_2 una transacción para transferir el 10% del saldo de la cuenta A a la cuenta B:

```
T2: Leer(A);  
    temp:= A*0.1  
    A:=A – temp  
    Escribir(A);  
    Leer(B)  
    B:=B + temp  
    Escribir(B).
```

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Sean los valores de $A=2000$ y $B=4000$.
- Supongamos que las dos transacciones se ejecutan de una en una
 - T1 primero y luego T2
 - T2 primero y luego T1

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

T1 -> T2

T1: Leer(A);
 A:=A - 1000
 Escribir(A);
 Leer(B)
 B:=B + 1000
 Escribir(B).

T2: Leer(A);
 temp:= A*0.1
 A:=A - temp
 Escribir(A);
 Leer(B)
 B:=B + temp
 Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
 Latino, Septiembre 2006

T2 -> T1

T2: Leer(A);
 temp:= A*0.1
 A:=A - temp
 Escribir(A);
 Leer(B)
 B:=B + temp
 Escribir(B).

T1: Leer(A);
 A:=A - 1000
 Escribir(A);
 Leer(B)
 B:=B + 1000
 Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
 Latino, Septiembre 2006

- Las secuencias descritas se les denomina planificaciones.
- Representan el orden cronológico en el cual se ejecutan las transacciones.
- Estas transacciones son secuenciales.
- Cada planificación consiste en una secuencia de instrucciones de varias transacciones, las instrucciones de una única transacción están juntas en dicha planificación.
- Para un conjunto de n transacciones existen $n!$ planificaciones secuenciales válidas distintas.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Concurrencia

- Se refiere al hecho de que los SDBD permiten que varias transacciones accedan a una misma BD a la vez.
- Se necesita entonces un mecanismo de control de concurrencia para asegurar que las transacciones concurrentes no interfieran entre sí.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Problemas de concurrencia

- Actualización perdida
 - Se pierde la actualización de una transacción debido a la actualización de otra transacción.
- Dependencia no confirmada (comprometida)
 - Permite que una transacción recupere o modifique una tupla que ha sido actualizada por otra transacción.
- Análisis inconsistente
 - Una transacción lee varios valores de la BD y otra transacción actualiza algunos de estos valores durante la ejecución de la primera transacción.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Actualización perdida

Time	A	B	X
t1		Begin_transaction	100
t2	Begin_transaction	read(X)	100
t3	read(X)	$X = X + 100$	100
t4	$X = X - 10$	write(X)	200
t5	write(X)	commit	90
t6	commit		90

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Dependencia no comprometida

Time	A	B	X
t1		Begin_transaction	100
t2		read(X)	100
t3		$X = X + 100$	100
t4	Begin_transaction	write(X)	200
t5	read(X)	...	200
t6	$X = X - 10$	roolback	100
t7	write(X)		190
t8	commit		190

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Análisis inconsistente

Time	A	B	X	Y	Z	Sum
t1		Begin_transaction	100	50	25	
t2	Begin_transaction	sum = 0	100	50	25	0
t3	read(X)	read(X)	100	50	25	0
t4	$X = X - 10$	sum = sum + X	100	50	25	100
t5	write(X)	read(Y)	90	50	25	100
t6	read(Z)	sum = sum + Y	90	50	25	150
t7	$Z = Z + 10$		90	50	25	150
t8	write(Z)		90	50	35	150
t9	commit	read(Z)	90	50	35	150
t10		sum = sum + Z	90	50	35	185
t11		commit	90	50	35	185
		175?				

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Al ejecutar concurrentemente varias transacciones, la planificación correspondiente no tiene porqué ser secuencial.
- Se pueden ejecutar una transacción durante un tiempo corto, luego ejecutar la segunda transacción durante un tiempo y cambiar luego a la primera y así sucesivamente hasta terminar las dos transacciones.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

T1: Leer(A);
A:=A - 1000
Escribir(A);

T2: Leer(A);
temp:= A*0.1
A:=A - temp
Escribir(A);

Leer(B)
B:=B + 1000
Escribir(B).

Leer(B)
B:=B + temp
Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- No todas las ejecuciones producen un estado consistente.
- Es tarea del sistema de base de datos (componente de control de concurrencia) asegurar que cualquier planificación que se ejecute lleva a la BD a un estado consistente.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

T1: Leer(A);
A:=A - 1000

T2: Leer(A);
temp:= A*0.1
A:=A - temp
Escribir(A);
Leer(B)

Escribir(A);
Leer(B)
B:=B + 1000
Escribir(B).

B:=B + temp
Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Se puede asegurar la consistencia de la BD en una ejecución concurrente:
 - Si se está seguro de que cualquier planificación que se ejecute tiene el mismo efecto que otra que se hubiese ejecutado sin concurrencia.
- La planificación debe ser equivalente a una planificación secuencial.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Seriabilidad

- Las transacciones individuales son tomadas como correctas.
- Es correcta la ejecución de una transacción a la vez en cualquier orden serial.
- Una ejecución intercalada es correcta cuando equivale a alguna ejecución serial, es decir es serializable.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Secuencialidad en cuanto a conflictos

- Sea P una planificación que tiene dos instrucciones consecutivas I_i y I_j pertenecientes a las transacciones T_i y T_j respectivamente.
- Si I_i y I_j se refieren a distintos elementos de datos se pueden intercambiar I_i y I_j sin afectar el resultado.
- Si I_i y I_j se refieren al mismo elemento X , el orden de los pasos es importante.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Sean dos operaciones $\text{Read}(X)$ y $\text{Write}(X)$, se deben considerar cuatro pasos:
 1. $I_i = \text{Leer}(X)$, $I_j = \text{Leer}(X)$, no importa el orden de I_i y I_j puesto que leen el mismo valor de X .
 2. $I_i = \text{Leer}(X)$, $I_j = \text{Escribir}(X)$, el orden es importante ¿porqué?
 3. $I_i = \text{Escribir}(X)$, $I_j = \text{Leer}(X)$, el orden es importante ¿porqué?
 4. $I_i = \text{Escribir}(X)$, $I_j = \text{Escribir}(X)$, no importa el orden, sin embargo la siguiente instrucción $\text{Read}(X)$ si se ve afectada ¿porqué?

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Se dice que I_i e I_j tienen conflicto si existen operaciones de diferentes transacciones sobre el mismo elemento de datos y al menos una de las operaciones es Escribir(X).

Leer(A); Escribir(A);	
	Leer(A); Escribir(A);
Leer(B) Escribir(B).	
	Leer(B) Escribir(B).

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Si I_i y I_j son instrucciones consecutivas de una planificación y de transacciones diferentes y además no están en conflicto, entonces el orden de I_i y I_j se puede intercambiar para obtener una nueva planificación P' .
- P' debe ser equivalente a P .

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Leer(A); Escribir(A);	
Leer(B)	Leer(A);
Escribir(B).	Escribir(A);
	Leer(B) Escribir(B).

- Intercambiar Leer(B) de T1 con Leer(A) de T2
- Intercambiar Escribir(B) de T1 con Escribir(A) de T2
- Intercambiar Escribir(B) de T1 con Leer(A) de T2

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

- Si una planificación P se puede transformar en otra P' por medio de una serie de intercambios de instrucciones no conflictivas, se dice que P y P' son equivalentes en cuanto a conflictos.
- Una planificación P es secuenciable en cuanto a conflictos si es equivalente en cuanto a conflictos a una planificación secuencial.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Protocolos basados en el bloqueo

- Una forma de asegurar la secuenciabilidad es exigir que el acceso a los elementos de datos se haga en exclusión mutua, esto es, mientras una transacción accede a un elemento ninguna transacción puede modificar dicho elemento.
- Métodos de bloqueo:
 - Compartido
 - Exclusivo

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Métodos de bloqueo

- Bloqueo compartido (C)
 - Si una transacción T_i obtiene un bloqueo en modo compartido sobre el elemento Q , entonces T_i puede leer Q pero no lo puede escribir.
- Bloqueo exclusivo (X)
 - Si la transacción T_i obtiene un bloqueo en modo exclusivo sobre el elemento Q , entonces T_i puede leer y escribir Q .

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Matriz de compatibilidad

	C	X
C	Si	No
X	No	No

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Ejemplo

- Sea T_1 una transacción para transferir \$1000 desde la cuenta B a la cuenta A:

T_1 : bloquear-X(B)
 Leer(B);
 $B := B - 1000$
 Escribir(B);
 desbloquear(B)
 bloquear-X(A)
 Leer(A)
 $B := A + 1000$
 Escribir(A).
 desbloquear(A)

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Ejemplo ...

- La transacción T2 visualiza la cantidad total de dinero de las cuentas A y B, es decir, la suma $A + B$:

T2: bloquear-C(A);
Leer(A);
desbloquear(A);
bloquear-C(B);
Leer(B);
desbloquear(B);
visualizar(A+B).

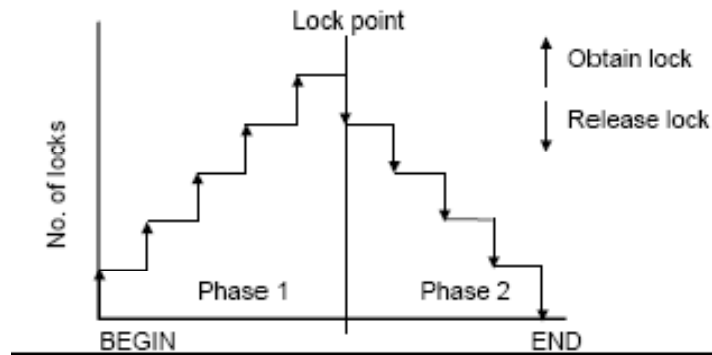
M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Protocolo de bloqueo de dos fases (2PC)

- Asegura la secuenciabilidad.
- Exige que cada transacción realice las peticiones de bloqueo y desbloqueo de dos fases:
 - Fase de crecimiento: Una transacción puede obtener bloqueos pero no puede liberarlos.
 - Fase de decrecimiento: Una transacción puede liberar bloqueos pero no puede obtener ninguno nuevo.

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Protocolo de bloqueo de dos fases (2PC)



M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Ejemplo

```
Bloquear-X(B);
Leer(B);
B:=B-1000
Escribir(B)
Bloquear-X(A);
Leer(A);
B:=A-1000
Escribir(A)
Desbloquear(B)
Desbloquear(A)
```

```
Bloquear-C(A);
Leer(A);
Bloquear-C(B);
Leer(B);
Visualizar(A+B)
Desbloquear(A)
Desbloquear(B)
```

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006

Interbloqueo (deadlock)

Bloquear-X(B);
Leer(B);
B:=B-1000
Escribir(B)

Bloquear-C(A)
Leer(A)
Bloquear-C(B)

Bloquear-X(A)

M.C. Luis R. Basto Díaz, Centro Educativo
Latino, Septiembre 2006