

TP1-T4 (Teoría Control de Concurrencia de Transacciones Parte I)

1. Cual la diferencia entre un SGBD Mono Usuario y un SGBD Multiusuario

Sistemas monousuario.

Son sistemas muy simples que solo permiten el acceso a un usuario cada vez, por lo que no se requiere ningún tipo de restricción o control en la gestión de los usuarios conectados. Este tipo de sistemas pueden basarse tanto en la monoprogramación como en la multiprogramación y suele ser principalmente ordenadores personales.

Sistemas multiusuario.

Son sistemas que a su vez se basan en sistemas multiprogramados, permitiendo el acceso de varios usuarios simultáneamente.

Los usuarios tienen la posibilidad de ejecutar varios programas al mismo tiempo, lo que permite elevar al máximo el rendimiento del procesador.

2. Explique el concepto de Multiprogramación en los Sistema Operativos

Se denomina multiprogramación a una técnica por la que dos o más procesos pueden alojarse en la memoria principal y ser ejecutados concurrentemente por el procesador o CPU.

3. Explique los conceptos de Concurrencia Intercalada y Paralela

Concurrencia significa que dos o más cálculos ocurren dentro del mismo período de tiempo, y generalmente hay algún tipo de dependencia entre ellos.

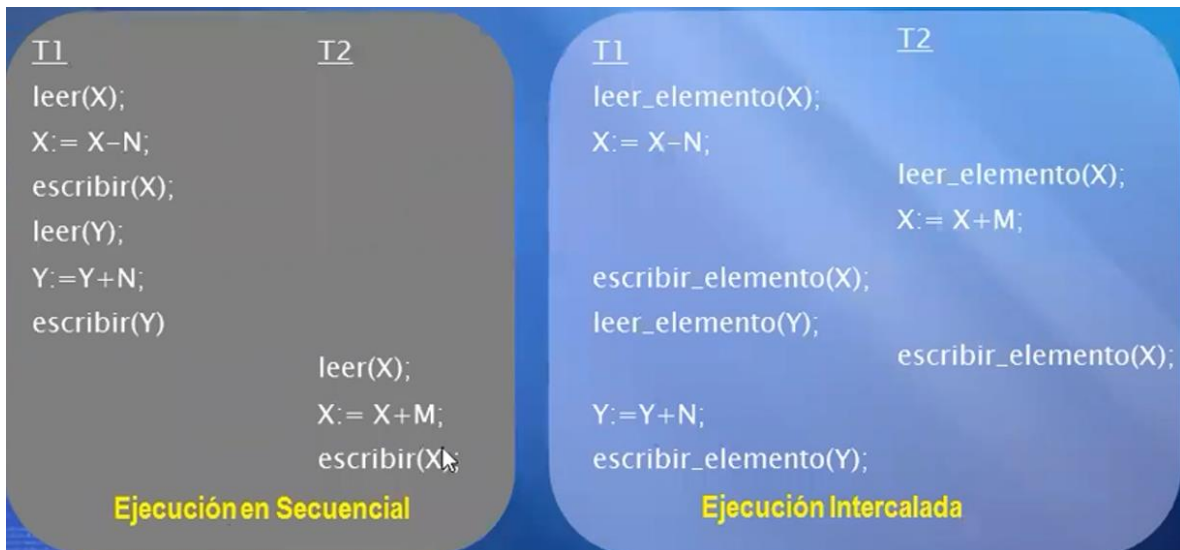
Paralelismo significa que dos o más cálculos ocurren simultáneamente.

4. Explique el concepto de Transacciones Concurrentes

Cuando dos o más transacciones se ejecutan al mismo tiempo y pueden modificar el mismo elemento de datos almacenados en la base de datos.

5. Dadas dos transacciones T1 y T2, explique las dos formas posibles que tiene el SGBD para ejecutar ambas transacciones.

T1	T2
leer (X);	leer(X);
X:= X-N;	X:= X+M;
escribir(X);	escribir(X);
leer(Y);	
Y:=Y+N;	
escribir(Y);	



6. Cites algunas razones por la cual es importante la ejecución concurrente de dos o más transacciones.

- Aumenta la productividad: el numero de transacciones ejecutadas por minuto.
- Aumenta la utilización de la CPU (menos tiempo ociosa) y Control del disco.
- Reduce el tiempo medio de respuesta de transacciones (las 'pequeñas' no esperan a las más 'grandes').

7. Las transacciones concurrentes deben controlar dos operaciones. ¿Cuáles son?

Primera fase: bloqueos.

Segunda: desbloqueos.

8. Defina el concepto de lectura de valor Fantasma en las Transacciones

Una lectura fantasma ocurre cuando, durante una transacción, se ejecutan dos consultas idénticas, y los resultados de la segunda no son iguales a los de la primera.

Esto puede ocurrir cuando no se realizan bloqueos de rango al realizar una operación SELECT ... WHERE.

9. Defina el concepto de lectura de valor Sucio en las Transacciones

sucede cuando una transacción hace un SELECT y obtiene un dato modificado por un UPDATE de otra transacción que no ha hecho commit. El dato es sucio porque todavía no ha sido confirmado y puede cambiar.

Por ejemplo: (1) un dato tiene un valor V0; (2) la transacción T1 lo actualiza a V1; (3) la transacción T2 lee el valor V1 del dato; (4) la transacción T1 hace un rollback, volviendo el dato al valor V0, y quedando sucio el dato contenido en la transacción T2.

10. Cite por lo menos 3 ejemplos que ocasiona ejecutar Transacciones Concurrentes.

Problemas de la Concurrency (ejecución intercalada)
i) Problemas de Actualización Perdida

Valores en la BD $X = 80, N = 5, M = 4, Y = 70$

T1	T2
$X = 80$ leer(X);	$X = 80$ leer(X);
$X = 80 - 5$ $X = X - N$; ($X = 75$)	$X = 75$ $X = X + M$; ($X = 84$)
$X = 80 + 4$	
escribir(X);	
$Y = 70$ leer(Y);	
$Y = 70 + 5$ $Y = Y + N$; ($Y = 75$)	
$Y = 75$ escribir(Y);	

~~$X = 84$~~ escribir(X);

El elemento $X = 84$ tiene un valor incorrecto debe ser $X = 79$ porque su actualización por T1 se 'perdió' (se sobrescribió)

Problemas de la Concurrency (ejecución intercalada)

ii) Problemas de la Actualización Temporal

La **actualización temporal** (o lectura sucia), T1 actualiza un elemento X de la BD y luego falla, pero **antes** de que se restaure el valor original de X, T2 tiene acceso al "valor temporal" de X

T1
leer(X);
X := X - N;
escribir(X);

T2

leer(X);
X := X + M;
escribir(X);

leer(Y);

...



T1 falla y debe devolver a X su antiguo valor, pero mientras, T2 ha leído el valor 'temporal' incorrecto de X (dato sucio)

Problemas de la Concurrency (ejecución intercalada)

iii) Problemas del Resumen Incorrecto

El **resumen incorrecto**, Otra transacción T3 calcula una función agregada de resumen sobre varios registros (suma las plazas reservadas para todos los vuelos), mientras otras transacciones, como T1, actualizan dichos registros: puede que T3 considere unos registros antes de ser actualizados y otros después

T1

leer(X);
X := X - N;
escribir(X);



leer(Y);
Y := Y + N;
escribir(Y);

T3

suma := 0;
leer(A);
suma := suma + A;

...

...

...

leer(X);
suma := suma + X;
leer(Y);
suma := suma + Y;

...

...

...

T3 lee X después de restar N, pero lee Y antes de sumar N, así que el resultado es un resumen incorrecto (discrepancia de N)

11. En los SGBD ¿Cual es función del Planificador de Transacciones?.

Regula el orden en el que se ejecutan las acciones de las diferentes transacciones.

Le llegan peticiones de lectura/escritura, y bien las ejecuta en los buffers o bien las retrasa

12. Que es el Protocolo de Control de Concurrency

Es el encargado de garantizar la atomicidad, el aislamiento y serialización de transacciones simultáneas.

13.Cuál es el objetivo del Protocolo de Control de Concurrency

Planificar las tracciones de forma que no ocurren interferencias entre ellas, y así evitar la aparición de los problemas.

No permitir intercalación de operaciones de varias transacciones.

14. Si los SGBD permiten ejecutar de manera intercalas las instrucciones de T1 y T2, cite por lo menos tres formas de ejecutar ambas transacciones.

T1	T2
leer (X);	leer(X);
X:= X-N;	X:= X+M;
escribir(X);	escribir(X);
leer(Y);	
Y:=Y+N;	
escribir(Y);	

15. Que es una Planificación de Transacciones y que condiciones debe cumplirse.

El planificador controla el orden en que el administrador de datos procesa los procedimientos de lectura y escritura sobre la base de datos. Cuando un planificador recibe este tipo de operaciones del envío del administrador de transacciones, tiene tres opciones: puede resolver la operación con la sencilla respuesta de salida a la misma, retardar la operación mediante el mantenimiento de la misma para acciones posteriores, o rechazar la operación, lo que significaría el rechazo de un aparte de la transacción incidente.

Esto deriva en una labor del planificador que desencadena la acción de abortar la transacción que produjo dicha operación cancelada. Asimismo, cualquier otra transacción ajena a esta, que hubiese incluido resultados modificados previos a la cancelación del proceso, tendría que ser de igual manera detenida y abortada.

16. La tabla 1 especifica la abreviación de las operaciones que participan en una Transacción, usando la tabla 1 se escriben las planificaciones P_A y P_B, escriba otras posibles planificaciones P_C, P_D y P_E. (El subíndice de cada operación indica a la transacción que pertenece)

P_A: l₁(X) ; e₁(X) ; l₁(Y) ; e₁(Y) ; c₁ ; l₂(X) ; e₂(X) ; c₂ ;

P_B: l₂(X) ; e₂(X) ; c₂ ; l₁(X) ; e₁(X) ; l₁(Y) ; e₁(Y) ; c₁ ;

P_C: l₂(X) ; l₁(X) ; e₁(X) ; l₁(Y) ; e₁(Y) ; c₁ ; e₂(X) ; c₂ ;

P_D: l₂(X) ; e₂(X) ; c₂ ; l₁(X) ; e₁(X) ; l₁(Y) ; e₁(Y) ; c₁ ; r

P_E: l₁(X) ; e₂(X) ; c₂ ; l₁(X) ; e₁(X) ; l₁(Y) ; e₁(Y) ; c₁ ;

operación	abreviatura
leer	l
escribir	e
commit	c
rollback	R

Tabla 1. Abreviación de las operaciones

T1	T2
leer (X);	leer(X);
X:= X-N;	X:= X+M;
escribir(X);	escribir(X);
leer(Y);	
Y:=Y+N;	
escribir(Y);	

17. Para las transacciones T1 y T2 ¿Cuál de las siguientes planificaciones son correctas y cuáles no?. Indicar en cada caso por qué.

P_A: l₁(X) ; e₁(X); l₁(Y); e₁(Y); c₁; l₂(X) ; e₂(X) ; c₂;

P_B: e₁(Y); l₁(X) ; l₁(Y); e₁(X) ; c₁; e₂(X) ; l₂(X) ; c₂;

P_C: l₂(X); c₂; l₁(X); e₁(X);e₁(Y); c₁;

P_D: l₂(X); e₂(X); c₂; l₁(X);e₁(X);l₁(Y); e₁(Y) ; c₁;

T1	T2
leer (X); X:= X-N; escribir(X); leer(Y); Y:=Y+N; escribir(Y);	leer(X); X:= X+M; escribir(X);

P_A: Es correcta por que sigue el orden de las transacciones en serie

P_B: Es incorrecta por que no se ejecutan las transacciones en orden

P_C: Es incorrecta por que no se ejecutan las transacciones en orden

P_D: Es correcta por que sigue el orden de las transacciones en serie

18. Defina el concepto de Planificación Serie

Es aquella en la que las operaciones de cada transacción se ejecutan consecutivamente son que se intercalen operaciones de otras transaccioes.

19. Defina el concepto de Planificación No Serie

Es aquella en la que las operaciones de un conjunto de transacciones concurrentes se ejecutan intercaladas, ejemplo:

20. Cual la diferencia entre Planificación Serie y Planificación No serie

La planificación en serie sus transacciones se ejecutan una a la vez, es mas segura pero es ineficiente, mas lenta.

La planificación no serie sus transacciones se ejecutan intercaladamente, es menos segura por que no todas las planificaciones pueden ser correctas, pero es más eficiente.

21. Para las transacciones T1 y T2 ¿Cuál de las siguientes planificaciones son Serie y cuáles no?. Indicar en cada caso por qué.

$P_A: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; c_2 ; l_1(Y) ; e_1(Y) ; c_1 ;$

No es serie por que se las operaciones de las transacciones se ejecutan intercaladas

$P_B: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; l_1(Y) ; c_2 ; e_1(Y) ; c_1 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_C: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; l_1(Y) ; e_1(Y) ; c_2 ; c_1 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_D: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; l_1(Y) ; e_1(Y) ; c_1 ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_E: l_1(X) ; e_1(X) ; l_2(X) ; l_1(Y) ; e_2(X) ; e_1(Y) ; c_1 ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_F: l_1(X) ; e_1(X) ; l_2(X) ; l_1(Y) ; e_1(Y) ; e_2(X) ; c_1 ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_G: l_1(X) ; e_1(X) ; l_2(X) ; l_1(Y) ; e_1(Y) ; c_1 ; e_2(X) ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_H: l_1(X) ; e_1(X) ; l_1(Y) ; l_2(X) ; e_1(Y) ; c_1 ; e_2(X) ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_I: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; l_2(X) ; c_1 ; e_2(X) ; c_2 ;$

No es serie por que las operaciones de las transacciones se ejecutan intercaladas

$P_J: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; c_1 ; l_2(X) ; e_2(X) ; c_2 ;$

Es serie por que las operaciones de cada transacciones se ejecutan consecutivamente

T1	T2
leer (X); $X := X - N$; escribir(X); leer(Y); $Y := Y + N$; escribir(Y);	leer(X); $X := X + M$; escribir(X);

22. Que es un Plan Equivalente

Es aquel que produce los mismos cambios o efectos en la base de datos, después de ejecutar 2 planes producen el mismo efecto.

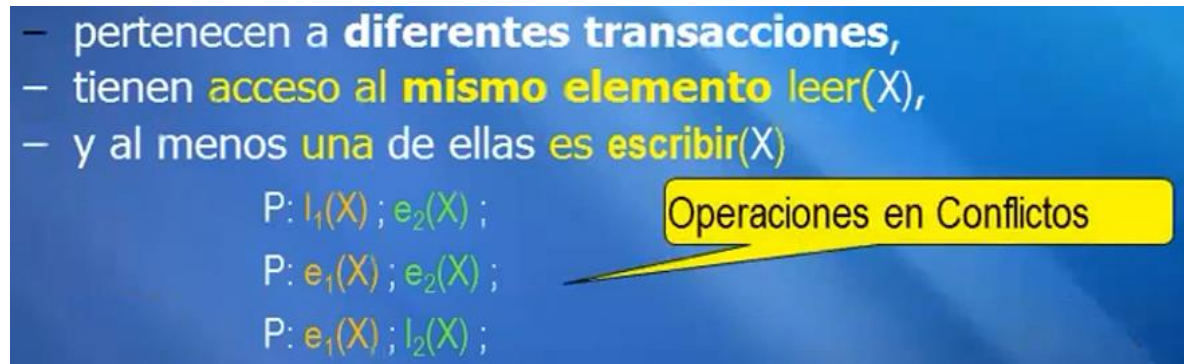
23. Defina el concepto de Planificación Serializable.

Una Planificación P (no serie) es serializable si es equivalente a alguna planificación serie de las mismas n transacciones.

24. Existen dos manera de definir la Equivalencia entre Transacciones ¿Cuáles son?

- Por conflictos
- De vistas

25. Explique ¿Cuándo dos operaciones de una Planificación están en conflicto?. Cite ejemplos.



El diagrama muestra tres condiciones para que dos operaciones estén en conflicto:

- pertenecen a **diferentes transacciones**,
- tienen **acceso al mismo elemento** leer(X),
- y al menos **una de ellas es escribir(X)**

Debajo de estas condiciones se listan tres ejemplos de operaciones:

- P: $l_1(X)$; $e_2(X)$;
- P: $e_1(X)$; $e_2(X)$;
- P: $e_1(X)$; $l_2(X)$;

Un recuadro amarillo con la etiqueta "Operaciones en Conflictos" apunta a los ejemplos.

26. Dada la siguientes Planificaciones, indicar cuales operaciones están en conflicto, para cada caso explicar por qué.

$P_A: l_1(X) ; l_2(X) ; e_1(X) ; l_1(Y) ; e_2(X) ; c_2 ; e_1(Y) ; c_1 ;$

Estan en conflicto por que acceden al mismo elemento leer(x)
y una de ellas es escribir(x).

$P_B: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; c_2 ; l_1(Y) ; e_1(Y) ; c_1 ;$

Estan en conflicto Estan en conflicto por que acceden al mismo elemento leer(x) y una de ellas es escribir(x).

27. ¿Cuándo un Plan No Serie es equivalentes por conflictos?

Cuando tienen el mismo acceso de leer a un determinado elemento x.

28. Dada la siguientes Planificación No Serie P_D , validar paso a paso si la misma es equivalente por Conflicto.

$P_D: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; c_2 ; l_1(Y) ; e_1(Y) ; c_1 ;$

NO, SE EJECUTAN LAS TRANSACCIONES CORRECTAMENTE

29. Describa los pasos del Algoritmo (usando grafos) que permite determinar si un Plan es serializable por conflictos.

Se intercambian cada dos operaciones de P consecutivas de transacciones distintas y sin conflicto, hasta obtener la planificación serie equivalente.

30. Dados los Planes P_A , P_B y P_C dibujar los grafos para probar la serializabilidad por conflictos usando el Algoritmo.

$P_A: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; c_1 ; l_2(X) ; e_2(X) ; c_2 ;$

$P_B: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; \overline{c_2} ; l_1(Y) ; e_1(Y) ; c_1 ;$

$P_{B1}: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; l_1(Y) ; \overline{c_2} ; e_1(Y) ; c_1 ;$

$P_{B2}: l_1(X) ; e_1(X) ; l_2(X) ; e_2(X) ; l_1(Y) ; e_1(Y) ; \overline{c_2} ; c_1 ;$

$P_{B3}: l_1(X) ; e_1(X) ; l_2(X) ; \overline{e_2(X)} ; l_1(Y) ; e_1(Y) ; c_1 ; c_2 ;$

$P_{B4}: l_1(X) ; e_1(X) ; l_2(X) ; l_1(Y) ; \overline{e_2(X)} ; e_1(Y) ; c_1 ; c_2 ;$

$P_{B5}: l_1(X) ; e_1(X) ; \overline{l_2(X)} ; l_1(Y) ; e_1(Y) ; e_2(X) ; c_1 ; c_2 ;$

$P_{B6}: l_1(X) ; e_1(X) ; l_1(Y) ; \overline{l_2(X)} ; e_1(Y) ; e_2(X) ; c_1 ; c_2 ;$

$P_{B7}: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; \overline{l_2(X)} ; e_2(X) ; c_1 ; c_2 ;$

$P_{B8}: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; \overline{l_2(X)} ; c_1 ; e_2(X) ; c_2 ;$

$P_{B9}: l_1(X) ; e_1(X) ; l_1(Y) ; e_1(Y) ; c_1 ; \overline{l_2(X)} ; e_2(X) ; c_2 ;$

$P_B: \text{Es serializable por conflictos}$

$P_C: l_1(X) ; l_2(X) ; e_1(X) ; e_2(X) ; \overline{c_2} ; l_1(Y) ; e_1(Y) ; c_1 ;$

$P_{C1}: l_1(X) ; l_2(X) ; e_1(X) ; \overline{e_2(X)} ; l_1(Y) ; c_2 ; e_1(Y) ; c_1 ;$

$P_{C2}: l_1(X) ; l_2(X) ; e_1(X) ; l_1(Y) ; e_2(X) ; \overline{c_2} ; e_1(Y) ; c_1 ;$

$P_{C3}: l_1(X) ; l_2(X) ; e_1(X) ; l_1(Y) ; e_2(X) ; e_1(Y) ; \overline{c_2} ; c_1 ;$

$P_{c4}: I_1(X) ; I_2(X) ; e_1(X) ; I_1(Y) ; \overline{e_2(X)} ; e_1(Y) ; c_1 ; c_2 ;$

$P_{c5}: I_1(X) ; I_2(X) ; e_1(X) ; I_1(Y) ; e_1(Y) ; \overline{e_2(X)} ; c_1 ; c_2 ;$

$P_{c6}: I_1(X) ; \overline{I_2(X)} ; e_1(X) ; I_1(Y) ; e_1(Y) ; c_1 ; e_2(X) ; c_2 ;$

P_{c7} : No es serializable por conflictos.