BASE DE DATOS

PRACTICA DE RECUPERACION ANTE FALLAS (LOGGING)

Ejercicio 1.
Dada la siguiente historia:

H= R1(Z) W1(U) **C1** W2(X) W2(Y) R3(U) W3(X) W2(Z) **C2** R3(Y) R4(Z) W3(Y) **C3** R4(U) W4(U) **C4**

Suponga que inicialmente todos los items tienen valor 0 y cada operación WRITE incrementa en 1 el valor anterior.

Escriba la secuencia de los registros de log correspondientes a la ejecución exitosa de H, según:

a) Undo logging con checkpointing quiescente (incluir un checkpoint en el log)
b) Redo logging con checkpointing no quiescente (incluir un checkpoint en el log tal que ocurra durante la ejecución de H)
c) Undo/Redo logging sin checkpointing

Ejercicio 2.
Dada la siguiente secuencia de registros en el *undo-log* escritos por el *log manager* correspondientes a la ejecución de tres transacciones T1 , T2 y T12, sobre los ítems A, B, C, D , E y R:

<START T1>; <T1, A, 10>; <START T2>; <T2, B, 20>; <START T12>
<T1, C, 30>; <T2, D, 40>; <COMMIT T2>; <T12,R,12> ; <T1, E, 50>;
<ABORT T12><COMMIT T1>

Describa las acciones que debería realizar el *recovery manager* -incluyendo los cambios tanto en los ítems en disco como en el log- si ocurre un *crash* y el último registro de log en disco es:

a) <START T2>
b) <COMMIT T2>
c) <T1, E, 50>
d) <COMMIT T1>

Ejercicio 3.
Idem ejercicio anterior pero para *redo logging.*

Ejercicio 4.
a) Dada la siguiente secuencia de registros en el *undo-log* generados por el *log manager* correspondientes a la ejecución de las transacciones T1, T2 , T3 y T4:

<START T1>; <T1, A, 8>; <START T2>; <START T4> <T2, B, 16>;
<START T3>; <T3, E, 24>; <T2, D, 32>; <T4,K,9> ;<T3, F, 40>;
<ABORT T4> ; <COMMIT T3>; <T2, G, 48>; <COMMIT T2>; <T1, C, 56>;
<COMMIT T1>

Describa las acciones que debería realizar el *recovery manager* -incluyendo los cambios tanto en los ítems en disco como en el log- si ocurre un *crash* y el último registro de log en disco es:

i) <START T3>
ii) <T1, C, 56>

b) Idem ejercicio anterior pero asumiendo *redo logging* y que el último registro de log en disco es:

i) <COMMIT T3>
ii) <COMMIT T1>

Ejercicio 5.
Dada la siguiente secuencia de registros *undo/redo-log* generados por el *log manager* correspondientes a la ejecución de cuatro transacciones T1, T2, T3 y T4:

<START T1>; <T1, A, 100, 110>; <START T2>; <T2, B, 200, 210>; <START T3>; <T1, C, 300, 310>; <T3, D, 400, 410>; <T2, E, 40, 41>; <T3, F, 500,510>; <COMMIT T3>; <COMMIT T2>; <START T4>; <T1, G, 600, 610>; <T4, H, 700, 710>; <COMMIT T1>; <COMMIT T4>

Describa las acciones que debería realizar el *recovery manager* -incluyendo los cambios tanto en los items en disco como en el log- si ocurre un *crash* y el último registro de log en disco es:

a) <START T3>
b) <COMMIT T3>
c) <T1, G, 600, 610>
d) <COMMIT T4>

Ejercicio 6.
Dada la siguiente secuencia de registros de *undo-log*:

<START T1>; <T1, A, 60>; <COMMIT T1>; <START T2>; <T2, A, 10>; <START T3>; <T3, B, 20>; <T2, C, 30>; <START T4>; <T3, D, 40>; <T4, F, 70>; <COMMIT T3>; <T2, E, 50>; COMMIT T2>; <T4, B, 80>; <COMMIT T4>

Suponga que se inicia *checkpoint* no-quiescente inmediatamente después de que cada uno de los siguientes registros de log han sido escritos en el *pool de buffers* de memoria:

a) <T1, A, 60>
b) <T2, A, 10>
c) <T3, B, 20>
d) <T3, D, 40>
e) <T2, E, 50>

Para cada uno indicar:

i. Cuándo el registro <END CKPT> es escrito.
ii. Para cada punto donde fuera posible que ocurriera un *crash*, hasta dónde se debería examinar el log para hallar todas las posibles transacciones incompletas.

Ejercicio 7.
Dados los mismos datos del ejercicio anterior pero ahora asumiendo *redo logging* indicar para cada punto de a) hasta e):

i. En qué punto el registro <END CKPT> podría ser escrito.
ii. Para cada punto donde fuera posible que ocurriera un *crash*, hasta dónde se debería examinar el log para hallar todas las posibles transacciones completas. Considere los dos casos posibles: que el registro <END CKPT> fuera escrito o no antes del *crash.*

Ejercicio 8.
Dada la siguiente secuencia de registros de *undo/redo-log*:

<START T1>; <T1, A, 60, 61>; <COMMIT T1>; <START T2>; <T2, A, 61, 62>; <START T3>; <T3, B, 20, 21>; <T2, C, 30, 31>; <START T4>; <T3, D, 40, 41>; <T4, F, 70, 71>; <COMMIT T3>; <T2, E, 50, 51>; COMMIT T2>; <T4, B, 21, 22>; <COMMIT T4>

Suponga que se inicia *checkpoint* no-quiescente inmediatamente después de que cada uno de los siguientes registros de log han sido escritos en el *pool de buffers* de memoria:

a) <T1, A, 60, 61>
b) <T2, A, 61, 62>
c) <T3, B, 20, 21>

d) <T3, D, 40, 41>
e) <T2, E, 50, 51>

Para cada uno indicar:

i. En qué punto el registro <END CKPT> podría ser escrito.
ii. Para cada punto donde fuera posible que ocurriera un *crash*, hasta dónde se debería examinar el log para hallar todas las posibles transacciones incompletas. Considere los dos casos posibles: que el registro <END CKPT> fuera escrito o no antes del *crash.*

Ejercicio 9.
Dada la siguiente secuencia de registros *undo/redo-log* con *nonquiescent checkpointing* generados por el *log manager* correspondientes a la ejecución de las transacciones T1 a T5:

1. < START T1>
2. < T1, A, 26, 33>
3. < START T2>
4. < START T3>
5. < T2, C, 51, 34>
6. < T3, E, 21, 35>
7. < T1, B, 25, 24>
8. < T1, A, 33, 20>
9. < T2, D, 18, 36>
10. < COMMIT T1>
11. < START T4>
12. < T4, F, 19, 37>
13. < T2, D, 36, 4>
14. < START CKPT (T2, T3, T4)>
15. < T2, C, 34, 10>
16. < T4, F, 37, 41>
17. < START T5>
18. < T3, E, 35, 52>
19. < T5, G, 27, 43>
20. < COMMIT T2>
21. < END CKPT>
22. < T5, G, 43, 28>
23. < T4, H, 26, 23>
24. < COMMIT T5>
25. < COMMIT T4>

Asuma que las entradas en el log son de la forma <transacción, item, valor anterior, valor nuevo>.

Para los siguientes posibles escenarios de fallas:

(1) El sistema crashea justo antes de que la línea 23 sea escrita en el disco.
(2) Idem anterior pero para la línea 24.
(3) Idem anterior pero para la línea 25.
(4) El sistema crashea inmediatamente después de que la línea 25 fuera escrita en el disco.

Completar la siguiente tabla con los valores de los items A, B, C, D, E, F, G y H en el disco después de una recuperación exitosa.

Escenario	**Item**							
	A	B	C	D	E	F	G	H
(1)								
(2)								
(3)								
(4)								

Ejercicio 10.

Parte (a):
Dada la siguiente secuencia de registros *undo/redo-log* con *nonquiescent checkpointing* generados por el *log manager* correspondientes a la ejecución de las siguientes transacciones:

< START T1>, < T1, A, 10, 20>, < T1, B, 0, 40>, < START T2>,
< T1, A, 20, 50>, < T2, C, 20, 30>, < COMMIT T1>, < START T3>,
< T3, D, 30, 60>, < T2, E, 25, 50>, < START CKPT (T2, T3)>, < T2, C, 30, 45>,
< COMMIT T2>, < START T4>, <START T6>, <T6,L,10,100>, < T4, F, 10, 80>,
<ABORT T6>, < COMMIT T3>, < END CKPT>, < T4, F, 80, 100>, < COMMIT T4>

Asuma que los registros en el log son de la forma <transacción, item, valor anterior, valor nuevo>. Para los siguientes posibles escenarios de fallas:

(1) El sistema crashea justo antes de que el registro < COMMIT T2> sea escrito en el disco.

(2) Idem anterior pero para el registro < START T4>.

(3) Idem anterior pero para el registro < COMMIT T4>.

Completar la siguiente tabla con los valores de los items A, B, C, D, E, F y L en el disco después de una recuperación exitosa.

	Item						
Escenario	A	B	C	D	E	F	L
(1)							
(2)							
(3)							
(4)							

Parte (b):
(1) Suponga que se utiliza *undo-logging*. Se produce un crash y los registros de log en disco son:

< START T1>, < T1, A, 5>, < START T2>, < T1, B, 1>, < COMMIT T1>,
< T2, B, 11>, < T2, C, 8>, < COMMIT T2>, < START T3>, < T3, A, 10>,
< START T4>, <T4, A, 11>, < T3, C, 7>, < T4, B, 22>

Las últimas cuatro acciones fueron el incremento de cada item en 1 (por ej. < T4, A, 11> fue la actualización de A a 12).

¿Cuantas combinaciones de valores diferentes de los items A, B y C podrían existir en disco al momento del crash?

(2) Suponga que se utiliza *undo/redo-logging*. Se produce un crash y los registros de log en disco son:

< START T1>, < T1, A, 5, 0>, < START T2>, < T1, B, 1, 2>, < COMMIT T1>,
< T2, B, 2, 3>, < START CKPT(T2)>, < T2, C, 8, 9>, < COMMIT T2>,
< END CKPT>, < START T3>, < T3, A, 0, 10>, < START T4>, < T4, A, 10, 11>,
< T3, C, 9, 7>, < T4, B, 3, 22>

Los registros < START CKPT()> y < END CKPT> indican el inicio y fin de un *checkpoint* no-quiescente.

¿Cuantas combinaciones de valores diferentes de los items A, B y C podrían existir en disco al momento del crash?