RECUPERACION

Ejercicios de parcial

1 Un SGBD implementa el algoritmo de recuperación UNDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);	08 (WRITE T3, X, 18);
02 (BEGIN, T2);	09 (WRITE T2, Z, 6);
03 (WRITE T1, X, 0);	10 (COMMIT, T2);
04 (WRITE T2, Y, 0);	11 (END CKPT);
05 (COMMIT, T1);	12 (WRITE T3, Z, 8);
06 (BEGIN CKPT, T2);	13 (BEGIN, T4);
07 (BEGIN, T3);	14 (WRITE T4, Y, 4);

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando hasta qué punto del archivo de log se deberá retroceder, y qué cambios deberán realizarse en disco y en el log.

Leyendo desde atras hacia adelante, se encuentra primero un END CKPT. Por lo tanto, si lo matcheamos con su BEGIN CKPT (BEGIN CKPT T2), vamos a tener que retorocedes hasta el inicio de la transacción activa mas antigua, que en este caso es T2 (linea 2).

Para el proceso de recuperación, se deberán deshacer todas las transacciones después del BEGIN CKPT que NO commitearon.

En este caso, hay que deshacer las transacciones T3 y T4. Se deberá reemplazar

- Y por 4
- Z por 8
- X por 18

Deshacemos T4, escribiendo Y=4. Hacemos flush de este dato a disco. Escribimos (ABORT, T4) en el log, y hacemos flush del log a disco. Deshacemos T3 escribiendo Z=8 y X=18. Hacemos flush de esos datos a disco. Escribimos (ABORT, T3) en el log, y hacemos flush del log a disco.

2 Un SGBD implementa el algoritmo de recuperación UNDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);	07 (BEGIN, T3);
02 (BEGIN, T2);	08 (WRITE T3, A, 8);
03 (WRITE T1, A, 13);	09 (WRITE T2, B, 15);
04 (WRITE T2, C, 7);	10 (BEGIN, T4);
05 (COMMIT, T1);	11 (WRITE T3, B, 8);
06 (BEGIN CKPT, T2);	12 (WRITE T4, C, 9);

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando hasta qué punto del archivo de log se deberá retroceder, y qué cambios deberán ser realizados en disco y en el archivo de log.

El primer CKPT que se encuentra es un BEGIN CKPT. Por lo tanto, debemos volver hacia atrás hasta el inicio de la transacción mas antigua del listado de activas. En este caso, es hasta el inicio de la T2.

Luego, voy a hacer el UNDO de todas las que no están committeadas. En este caso seria un UNDO de T2, T3 y T4. Por estas:

- Reescribo 7 en C
- Reescribo 15 en B

Escribimos (ABORT, T2) y hacemos el flush a disco.

Deshacemos también T3 y T4, que no commitearon:

- Escribo C=9

Hacemos flush de este dato a disco, escribimos (ABORT T4) y hacemos flush del log a disco

Para deshacer T3 escribimos B=8 y A=18. Hacemos flush de estos datos a disco. Luego hacemos (ABORT T3) y hacemos flush del log a disco.

3 Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);	09 (BEGIN, T4);
02 (WRITE, T1, A, 6);	10 (WRITE, T3, A, 12);
03 (BEGIN, T2);	11 (COMMIT T2);
04 (WRITE, T2, B, 12);	12 (WRITE, T3, C, 21);
05 (WRITE, T2, C, 3);	13 (END CKPT);
06 (COMMIT, T1);	14 (WRITE, T4, B, 24);
07 (BEGIN CKPT, T2);	15 (COMMIT, T3);
08 (BEGIN, T3);	

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

El primer checkpoint que se encuentra es un END CKPT. Se debe retrocedes hasta el BEGIN de la transacción mas antigua que figure en el BEGIN CKPT.

En este caso, el BEGIN CKPT registra a la T2, así que debo retroceder hasta la línea 03 donde esta el begin de T2.

Luego, voy a hacer un REDO de las transacciones que SI commitearon. En este caso, seria T2 y T3.

Voy a rehacer B=12, C=3, A=12, C=21. Luego, voy a abortar las transacciones que NO commitearon, en este caso seria ABORT T4. Hago un flush de los valores a disco y un flush del log a disco.

4 Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);
02 (WRITE, T1, A, 5, 3);
03 (BEGIN, T2);
04 (WRITE, T2, B, 4, 8);
05 (WRITE, T1, C, 3, 2);
06 (BEGIN, T3);
07 (WRITE, T3, D, 15, 12);
08 (COMMIT, T3);
09 (BEGIN CKPT, {T1, T2});
10 (BEGIN, T4);
11 (WRITE, T2, D, 12, 10);
12 (WRITE, T4, E, 8, 18);
13 (COMMIT, T2);
14 (WRITE, T1, B, 8, 3);
15 (END CKPT);
16 (COMMIT, T1);
17 (WRITE, T4, C, 2, 1);

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

Voy a tener que volver hasta el inicio de la transacción mas antigua registrada en el (BEGIN CKPT T1, T2). Voy a volver hasta el inicio del log en la liena 01, donde tengo el BEGIN T1.

Voy a tener que deshacer todas las transacciones que NO commitearon, en este caso T4.

Para ello reescribo C=2, E=8. Hago un flush de esos datos a disco. Luego, escribo ABORT(T4) en el log, y hago un flush del log a disco.

Ahora, aplico REDO sobre las transacciones que SI commitearon. En este caso, T1 y T2 **porque solo miro las transacciones marcadas como activas en el BEGIN CKPT rehago las operaciones de las transacciones desde el checkpoint en adelante**

Reescribo los valores modificados por T1: B=3. Hago un flush de esos datos a disco.

Reescribo los valores modificados por T2:D=10. Hago un flush de esos datos a disco

La palabra final sobre B la tiene T1 porque es la última que opera.

5 El gestor de log de una base de datos utiliza un log de tipo REDO. Indique si cada una de las siguientes afirmaciones sobre el funcionamiento del mismo es verdadera (V) o falsa (F), justificando su respuesta.

a) El registro de log correspondiente a un write item(X) debe ser volcado a disco después de volcar el nuevo valor de X modificado por dicha instrucción a disco.

b) El registro de log correspondiente al commit de una transacción Ti debe ser volcado a disco después de volcar el nuevo valor de todo ítem X modificado por Ti a disco.

c) Un checkpoint activo recién podrá terminarse (es decir, recién se podr´a escribir el END CKPT en el log) una vez que todas las transacciones que estaban activas al inicio del checkpoint hayan terminado y sus datos hayan sido enviados a disco.

d) En caso de recuperación, si se debe abortar una transacción Ti que no había terminado, entonces antes de escribir el registro de log correspondiente al abort de Ti deberán volcarse a disco los valores anteriores vold de cada ítem que había sido modificado por Ti.

a) FALSO: Esto no respeta las reglas básicas del gestor, WAL y FLC. Antes del commit se flushea el log a disco, y solo después del commit se puede flushear los writes a disco, por o que si o si se va a escribir el log antes del write.

b) FALSO: cuando la transacción hace commit, se escribe el commit en el log y se hace flush del log a disco por la regla FLC. Recién entonces se escribe el nuevo valor en disco.

c) FALSO: una transacción puede ser marcada como activa en el BEGIN CKPT correspondiente al END CKPT y que su commit (es decir su finalización) se encuentre después del END CKPT. El END CKPT se hace cuando se flushearon a disco todas las transacciones que commitearon ANTES del BEGIN CKPT

d) FALSO: los valores se vuelcan después del commit, y si aborto nunca hubo commit, por lo que nunca se volcaron los nuevos valores. Como es un algoritmo REDO, se deben volcar a disco los valores nuevos vnew de las transacciones que SI terminaron (las que commitearon)

6 Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Indique si las siguientes afirmaciones sobre el funcionamiento del algoritmo son verdaderas o falsas, justificando su respuesta:

(a) Los ítems de datos modificados por una transacción Ti deben ser flusheados a disco antes de escribir (COMMIT, Ti) en el archivo de log.

(b) Cuando se modifica un ítem de datos X es necesario registrar en el log tanto su valor anterior como su nuevo valor.

(c) Cuando se modifica un ítem de datos X es obligatorio flushear el nuevo valor de X a disco antes de flushear el registro de log correspondiente a disco.

(d) Si el sistema se reinicia y el algoritmo detecta que una transacción Ti no había llegado a commitear, se debe escribir (ABORT, Ti) en el archivo de log y flushearlo a disco antes de deshacer las modificaciones realizadas por Ti en disco.

a. FALSO: los ítems modificados pueden ser guardados en disco antes o después de hacer commit.
b. VERDADERO: se registra el valor viejo y el valor nuevo, porque en caso de una falla, se deberá rehacer la operación con el valor nuevo si la Tx committeo, o se deberá deshacer la operación, pisando con el valor viejo si no committeo.
c. FALSO. el registro debe ser escrito en el log en disco antes de escribir el nuevo valor del dato en disco. **Regla WAL**
d. FALSO: primero se deshacen los cambios realizados por la transacción y se hace flush de esos datos a disco. Luego se añade al log el ABORT de la transacción y se hace flush de eso a disco.

7 Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);
02 (WRITE T1, A, 10, 15);
03 (BEGIN, T2);
04 (WRITE T2, B, 20, 40);
05 (WRITE T1, C, 30, 35);
06 (BEGIN CKPT, T1, T2);
07 (WRITE T2, D, 40, 50);
08 (COMMIT, T2);
09 (WRITE T1, B, 40, 60);
10 (BEGIN, T3);
11 (WRITE T3, E, 15, 30);
12 (END CKPT);
13 (WRITE T3, D, 50, 15);

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

Dado que en BEGIN CKPT se registran como activas las transacciones T1 y T2, se deberá retroceder hasta la línea 01 donde comienza T1.

Primero realizamos el UNDO de las transacciones que NO commitearon, en este caso T1 y T3:

Reescribo los datos D=50, E=15. Hago flush de los datos a disco, y el ABORT(T3) al log. Luego el flush del log a disco

Reescribo los datos B=40, C=30, A=10. Hago flush de los datos a disco, y el ABORT(T1) al log. Luego el flush del log a disco

Ahora rehago la T2, que SI committeo. Reescribo los datos D=50