

## 2 Concurrencia y recuperabilidad

**Criterio de aprobación:** a) Debe tener bien el ítem iii) y alguno de los dos primeros. b) Debe tener al menos un ítem bien. c) Debe tener al menos un ítem bien.

a) Se tienen las transacciones  $T_1, T_2, T_3$  y la siguiente historia:

$$H = rl_1(A), rl_2(B), rl_3(B), wl_1(B), u_2(B), u_3(B), rl_2(A), u_1(A), u_1(B), u_2(A), wl_3(A), u_3(A), c_1, c_2, c_3$$

Se pide responder **justificando**:

- i) La historia  $H$ , ¿es legal?. En caso que no lo sea, construir  $H'$  realizando una cantidad minimal de cambios moviendo las operaciones de modo que sea legal. Justifique.
- ii) La historia legal del punto i) ¿es serializable? Justifique.
- iii) Analizar la historia con respecto a los criterios de recuperabilidad (RC, ACA y ST). Justifique.
- iv) ¿La historia es **2PL**? En caso de no serlo, analice y escriba si es posible algún cambio en las transacciones para que lo sea Justifique

b) Sean  $T_1, T_2, T_3$ . El planificador posee la siguiente historia:

$$H = st_1, st_3, st_2, r_1(X), r_2(Z), w_1(Z), w_3(X), r_3(Z), w_2(Y), c_2, w_3(Y), c_3$$

Los valores iniciales de  $X, Y$  y  $Z$  son 0 (cero). Suponiendo las siguientes escrituras:

$$T_1 : Z = 400$$

$$T_2 : Y = 50$$

$$T_3 : X = 300, Y = 150$$

Analice la historia e indique que sucede si se utiliza un protocolo de timestamp sin multiversion y con multiversion. Indique además como queda el valor de cada variable luego de finalizada la historia

c) Se dispone del siguiente undo log con checkpoint no-quietescente:

1. < START  $T_2$  >
2. <  $T_2, Z, 20$  >
3. < START  $T_1$  >
4. <  $T_1, Y, 3$  >
5. <  $T_1, X, 8$  >
6. < START  $T_3$  >
7. < COMMIT  $T_1$  >
8. <  $T_3, X, 10$  >
9. < COMMIT  $T_2$  >
10. < START CKPT( $T_3$ ) >
11. < START  $T_4$  >
12. <  $T_4, Y, 7$  >
13. < COMMIT  $T_3$  >
14. < END CKPT >
15. <  $T_4, Z, 1$  >
16. < ABORT  $T_4$  >

Responda, **justificando**:

- i) Suponga un *crash* luego del paso 15 (antes del 16). Describa que hace el *recovery manager* explicando como queda el log al finalizar y que paso con los ítems  $X, Y$  y  $Z$ .
- ii) Suponga un *crash* luego del paso 16. Describa que hace el *recovery manager* explicando como queda el log al finalizar y que paso con los ítems  $X, Y$  y  $Z$ .