

## Segundo Examen Parcial

CI-0127 Bases de Datos

Prof. Alexandra Martínez Porras

## Fórmula A

Total de puntos: 100

16 de julio 2019

Nombre:

Carné:

### Sección A [72 pts.]

Todas las preguntas de esta sección son de selección única. Marque su respuesta con una [X] sobre la letra correspondiente. Si se equivoca, tache totalmente la marca anterior y escriba "No" a la par. De no estar clara cuál es su respuesta, la pregunta se calificará como incorrecta.

1. [3 pts.] Dado el *schedule* de transacciones  $S_1$ :  $R_3(A)$ ;  $R_2(A)$ ;  $W_3(A)$ ;  $R_3(B)$ ;  $R_1(A)$ ;  $R_2(B)$ ;  $W_2(B)$ ;  $W_1(A)$ ;  $W_3(B)$ , indique si el grafo de precedencia contiene ciclos o no. Si hay ciclos, indique todos los ciclos. Si no, indique el *schedule* serial equivalente.
  - a) Sí tiene ciclos. Ciclos:  $T_2 \rightarrow T_1$ ;  $T_1 \rightarrow T_3$ ;  $T_3 \rightarrow T_2$ .
  - b) Sí tiene ciclos. Ciclos:  $T_2 \rightarrow T_3$ ;  $T_3 \rightarrow T_2$ .
  - c) No tiene ciclos. *Schedule* serial equivalente:  $T_2$ ;  $T_3$ ;  $T_1$ .
  - d) No tiene ciclos. *Schedule* serial equivalente:  $T_3$ ;  $T_2$ ;  $T_1$ .
  - e) Ninguna de las anteriores.
2. [3 pts.] Dado el *schedule* de transacciones  $S_2$ :  $L_1(A)$ ;  $R_1(A)$ ;  $L_3(B)$ ;  $W_3(B)$ ;  $U_1(A)$ ;  $L_2(B)$ ;  $R_2(B)$ ;  $U_2(B)$ ;  $U_3(B)$ ;  $L_3(C)$ ;  $W_3(C)$ ;  $U_3(C)$ , indique si las transacciones son bien formadas y si el *schedule* es legal de acuerdo al protocolo de *locking simple*.
  - a) Todas las transacciones están bien formadas y el *schedule* es legal.
  - b) Todas las transacciones están bien formadas pero el *schedule* no es legal.
  - c) No todas las transacciones están bien formadas pero el *schedule* sí es legal.
  - d) No todas las transacciones están bien formadas y el *schedule* no es legal.
  - a) Ninguna de las anteriores.
3. [3 pts.] Dado el *schedule* de transacciones  $S_3$ :  $L_1(A)$ ;  $R_1(A)$ ;  $L_3(B)$ ;  $R_3(B)$ ;  $L_3(C)$ ;  $W_3(B)$ ;  $U_3(B)$ ;  $L_2(B)$ ;  $R_3(C)$ ;  $R_2(B)$ ;  $U_2(B)$ ;  $U_3(C)$ , indique si las transacciones son bien formadas y si el *schedule* es legal de acuerdo al protocolo *two phase locking* (2PL).
  - a) Todas las transacciones están bien formadas y el *schedule* es legal.
  - b) Todas las transacciones están bien formadas pero el *schedule* no es legal.
  - c) No todas las transacciones están bien formadas pero el *schedule* sí es legal.
  - d) No todas las transacciones están bien formadas y el *schedule* no es legal.
  - a) Ninguna de las anteriores.

4. [3 pts.] ¿Cuáles de las siguientes afirmaciones son verdaderas?
- I. El protocolo de *locking S-X* (compartido-exclusivo) permite a dos transacciones distintas tener un *lock* de lectura ( $L_S$ ) sobre el mismo elemento de datos al mismo tiempo.
  - II. El 2PL-Conservador previene los *deadlocks* entre transacciones.
  - III. El 2PL-Estricto previene los *deadlocks* entre transacciones.
  - IV. El *waits-for graph* no permite detectar *deadlocks*.
- a) I
  - b) I y II
  - c) I y III
  - d) II y IV
  - e) Todas
  - f) Ninguna
5. [3 pts.] Bajo el esquema de *update locks*, se permite:
- a) Que una transacción que posee un *lock* de *update* ( $L_U$ ) sobre un elemento de datos  $A$  obtenga un *lock* exclusivo ( $L_X$ ) sobre ese mismo elemento de datos sin necesidad de liberar el *lock* actual  $L_U$ .
  - b) Que una transacción que posee un *lock* de lectura ( $L_S$ ) sobre un elemento de datos  $A$  obtenga un *lock* exclusivo ( $L_X$ ) sobre ese mismo elemento de datos sin necesidad de liberar el *lock* actual  $L_S$ .
  - c) Que una transacción que posee un *lock* de lectura ( $L_S$ ) sobre un elemento de datos  $A$  obtenga un *lock* de *update* ( $L_U$ ) sobre ese mismo elemento de datos sin necesidad de liberar el *lock* actual  $L_S$ .
  - d) Que una transacción que posee un *lock* de escritura ( $L_X$ ) sobre un elemento de datos  $A$  obtenga un *lock* de *update* ( $L_U$ ) sobre ese mismo elemento de datos sin necesidad de liberar el *lock* actual  $L_X$ .
  - e) (a) y (c).
  - f) (b) y (d).
  - g) Ninguna de las anteriores.
6. [3 pts.] Bajo el esquema de *wait-die* y asumiendo que  $TS(T_1) < TS(T_2)$ , indique qué ocurriría en el siguiente caso (la operación subrayada es una solicitud pendiente de aprobación):
- $T_1$ :  $L(X)$ ;  $R(X)$ ;  $W(X)$ ;  $L(Z)$  ...  
 $T_2$ :  $L(Z)$ ;  $W(Z)$ ;
- a) La transacción  $T_2$  aborta y la transacción  $T_1$  continúa.
  - b) La transacción  $T_1$  aborta y la transacción  $T_2$  continúa.
  - c) La transacción  $T_1$  espera por el *lock* y la transacción  $T_2$  continúa.
  - d) La transacción  $T_2$  espera por el *lock* y la transacción  $T_1$  continúa.
  - e) Ambas transacciones abortan.
  - f) Ninguna de las anteriores.

7. [3 pts.] Bajo el esquema de *wound-wait* y asumiendo que  $TS(T_1) < TS(T_2)$ , indique qué ocurriría en el siguiente caso (la operación subrayada es una solicitud pendiente de aprobación):

$T_1$ : L(X); R(X); W(X); L(Y) ...

$T_2$ : L(Y); L(V)

- a) La transacción  $T_1$  aborta y la transacción  $T_2$  continúa.
  - b) La transacción  $T_2$  aborta y la transacción  $T_1$  continúa.
  - c) La transacción  $T_1$  espera por el *lock* y la transacción  $T_2$  continúa.
  - d) La transacción  $T_2$  espera por el *lock* y la transacción  $T_1$  continúa.
  - e) Ambas transacciones abortan.
  - f) Ninguna de las anteriores.
8. [3 pts.] Los problemas de (i) *dirty read* (lectura sucia) y (ii) tuplas fantasma son posibles bajo los siguientes niveles de aislamiento:
- a) (i): read committed, (ii): repeatable read.
  - b) (i): read uncommitted, (ii): serializable.
  - c) (i): serializable, (ii): read committed.
  - d) (i): repeatable read, (ii): read uncommitted.
  - e) Ninguna de las anteriores
9. [3 pts.] El orden  $p$  un árbol B+ representa:
- a) La cantidad máxima de punteros que puede tener un nodo interno.
  - b) La cantidad mínima de punteros que debe tener un nodo hoja.
  - c) La cantidad mínima de llaves de búsqueda que debe tener el nodo raíz.
  - d) La cantidad de niveles que tiene el árbol.
  - e) Ninguna de las anteriores.

10. [3 pts.] Suponga que un cierto sistema de base de datos está utilizando una bitácora bajo el esquema de recuperación “rehacer” (*redo*). Asuma que la bitácora de la base de datos contiene lo siguiente:

Step	Action	Log
1)		<START T>
2)	READ(A,t)	
3)	t := t*2	
4)	WRITE(A,t)	<T, A, 16>
5)	READ(B,t)	
6)	t := t*2	
7)	WRITE(B,t)	<T, B, 16>
8)		<COMMIT T>
9)	FLUSH LOG	
10)	OUTPUT(A)	
11)	OUTPUT(B)	

¿Que pasaría si hubiese ocurrido un fallo grave (*crash*) en el sistema justo después del paso 9 pero antes del 10?

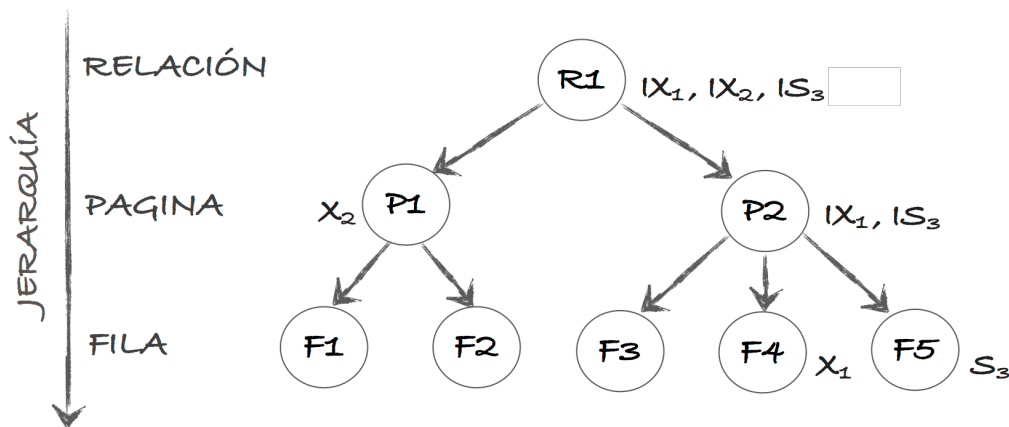
- Tendríamos que abortar la transacción T, escribiendo un registro <ABORT T> en la bitácora y forzando su escritura en disco.
  - Tendríamos que rehacer los cambios hechos por la transacción T, en el siguiente orden: A=16 y B=16, y forzar su escritura en disco.
  - No tendríamos que hacer nada puesto que la transacción T ya hizo COMMIT.
  - Ninguna de las anteriores.
11. [3 pts.] Suponga que un cierto sistema de base de datos está utilizando una bitácora bajo el esquema de recuperación “deshacer” (*undo*). Asuma que la bitácora de la base de datos contiene lo siguiente:

Step	Action	Log
1)		<START T>
2)	READ(A,t)	
3)	t := t*2	
4)	WRITE(A,t)	<T, A, 8>
5)	READ(B,t)	
6)	t := t*2	
7)	WRITE(B,t)	<T, B, 8>
8)	FLUSH LOG	
9)	OUTPUT(A)	
10)	OUTPUT(B)	
11)		<COMMIT T>
12)	FLUSH LOG	

¿Que pasaría si hubiese ocurrido un fallo grave (*crash*) en el sistema justo después del paso 9 pero antes del 10?

- Tendríamos que deshacer los cambios hechos por la transacción T, en el siguiente orden: B=8 y A=8, y forzar su escritura en disco.
- Sólo tendríamos que deshacer el cambio hecho por la transacción T al ítem A, ya que el cambio al ítem B nunca se escribió en disco.
- Tendríamos que abortar la transacción T, escribiendo un registro <ABORT T> en la bitácora y forzando su escritura en disco.
- No tendríamos que hacer nada puesto que la transacción T no ha hecho COMMIT aún.
- (a) y (c).
- (b) y (c).
- Ninguna de las anteriores.

12. [3 pts.] Bajo el esquema de *locks* de múltiple granularidad (*warning locks*) y asumiendo que el estado de los *locks* adquiridos por las transacciones sobre los elementos de datos es el que se muestra en la figura de abajo, indique qué ocurriría en cada uno de los casos I y II.



- Una nueva transacción  $T_4$  solicita *locks*  $IS(R1)$ ,  $IS(P2)$  y  $S(F5)$ .
- Una nueva transacción  $T_5$  solicita *locks*  $IX(R1)$ ,  $IX(P1)$  y  $X(F1)$ .

- Ambas  $T_4$  y  $T_5$  obtienen todos los *locks* que solicitaron.
- Ambas  $T_4$  y  $T_5$  quedan en espera del primer *lock* que solicitaron.
- $T_4$  obtiene todos los *locks*, y  $T_5$  obtiene  $IX(R1)$  pero queda en espera por  $IX(P1)$ .
- $T_4$  obtiene  $IS(R1)$  pero queda en espera por  $IS(P2)$ , y  $T_5$  obtiene  $IX(R1)$  pero queda en espera por  $IX(P1)$ .
- $T_4$  obtiene  $IS(R1)$  pero queda en espera por  $IS(P2)$ , y  $T_5$  queda en espera por  $IX(R1)$ .
- Ninguna de las anteriores.

13. [3 pts.] Una dependencia funcional  $A \rightarrow B$  se cumple en la relación R si...
- Para todas las parejas posibles de tuplas  $t_1$  y  $t_2$  donde  $t_1[A] = t_2[A]$ , se cumple que  $t_1[B] = t_2[B]$ .
  - Para todas las parejas posibles de tuplas  $t_1$  y  $t_2$  donde  $t_1[B] = t_2[B]$ , se cumple que  $t_1[A] = t_2[A]$ .
  - Existe una pareja de tuplas  $t_1$  y  $t_2$  con  $t_1[A] = t_2[A]$  para la cual se cumple que  $t_1[B] = t_2[B]$ .
  - Existe una pareja de tuplas  $t_1$  y  $t_2$  con  $t_1[B] = t_2[B]$  para la cual se cumple que  $t_1[A] = t_2[A]$ .
  - Ninguna de las anteriores.
14. [3 pts.] Dado el esquema de relación  $M(A, B, C, D, E, F)$  y su conjunto de dependencias funcionales  $\{A \rightarrow D, B \rightarrow AF, D \rightarrow E, DB \rightarrow C\}$ , ¿cuál es la clausura de  $\{B\}$ ?
- $\{B\}^+ = \{A, B, F\}$
  - $\{B\}^+ = \{A, B, D, F\}$
  - $\{B\}^+ = \{A, B, D, E, F\}$
  - $\{B\}^+ = \{A, B, C, D, E, F\}$
  - Ninguna de las anteriores
15. [3 pts.] ¿Cuál de las siguientes afirmaciones es falsa?
- Cuando normalizamos una relación a Forma Normal Boyce-Codd podrían perderse dependencias funcionales.
  - Si todos los atributos de una relación pertenecen a alguna llave candidata, la relación está al menos en Tercera Forma Normal.
  - Si el lado derecho de todas las dependencias funcionales de una relación está formado por atributos no-primos, la relación está en Forma Normal Boyce-Codd.
  - Si algún atributo no-primo de una relación depende parcialmente de alguna llave candidata, la relación incumple la Segunda Forma Normal.
  - Ninguna de las anteriores.
16. [3 pts.] Dado el esquema de relación  $Q(A, B, C, D)$ , su conjunto de dependencias funcionales  $\{A \rightarrow D, CB \rightarrow A, D \rightarrow C\}$  y sus llaves candidatas  $\{A, B\}$ ,  $\{B, C\}$  y  $\{B, D\}$ , ¿en qué forma normal está Q?
- 1FN
  - 2FN
  - 3FN
  - FNBC
17. [3 pts.] Dado el esquema  $R(A, B, C, D, E)$ , su conjunto de dependencias funcionales  $\{AC \rightarrow BE, BC \rightarrow ADE, E \rightarrow D\}$  y sus llaves candidatas  $\{A, C\}$  y  $\{B, C\}$ , ¿en qué forma normal está R?
- 1FN
  - 2FN
  - 3FN
  - FNBC

18. [3 pts.] Dado el esquema de relación  $S(A, B, C, D, E)$ , su conjunto de dependencias funcionales  $\{B \rightarrow A, BC \rightarrow DE, D \rightarrow C, E \rightarrow BCD\}$  y sus llaves candidatas  $\{B,C\}$ ,  $\{B,D\}$  y  $\{E\}$ , ¿en qué forma normal está  $S$ ?
- a) 1FN                                      b) 2FN                                      c) 3FN                                      d) FNBC
19. [3 pts.] Dado el esquema de relación  $T(A, B, C, D, E, F)$ , su conjunto de dependencias funcionales  $\{A \rightarrow CF, C \rightarrow D, B \rightarrow E\}$  y su llave  $\{AB\}$ , ¿cuál es el resultado de normalizar  $T$  hasta **formal normal Boyce-Codd**?
- a)  $T_1(\underline{B}, E)$  donde  $B \rightarrow E$ ,  $T_2(\underline{A}, B)$  donde  $A$  es FK( $T_4$ ) y  $B$  es FK( $T_1$ ), ...  
 $T_3(\underline{C}, D)$  donde  $C \rightarrow D$  y  $T_4(\underline{A}, C, F)$  donde  $A \rightarrow CF$  y  $C$  es FK( $T_3$ )
- b)  $T_1(\underline{A}, C, F)$  donde  $A \rightarrow CF$  y  $C$  es FK( $T_2$ ),  $T_2(\underline{C}, D)$  donde  $C \rightarrow D$ , y  $T_3(\underline{B}, E)$  donde  $B \rightarrow E$
- c)  $T_1(\underline{A}, F)$  donde  $A \rightarrow F$ ,  $T_2(\underline{C}, D)$  donde  $C \rightarrow D$ ,  $T_3(\underline{B}, E)$  donde  $B \rightarrow E$  y ...  
 $T_4(\underline{A}, B, C)$  donde  $A$  es FK( $T_1$ ),  $B$  es FK( $T_3$ ) y  $C$  es FK( $T_2$ )
- d)  $T_1(\underline{B}, E)$  donde  $B \rightarrow E$ ,  $T_2(\underline{A}, C, D, F)$  donde  $A \rightarrow CDF$ , y  $T_3(\underline{A}, B)$  donde  $A$  es FK( $T_2$ ) y  $B$  es FK( $T_1$ )
- e) Ninguna de las anteriores
20. [3 pts.] Asumiendo que se tiene un archivo de datos ordenado con 100.000 registros, donde el tamaño del bloque es 2.048 bytes, el tamaño de cada registro es 90 bytes, los registros son de tamaño fijo y la organización de registros en bloques es no-expandible, ¿cuál es el factor de bloque?
- a) 22                                      b) 23                                      c) 1.111                                      d) 1.112                                      e) ninguno
21. [3 pts.] Asumiendo que se tiene un archivo de datos ordenado con 50.000 registros, donde el tamaño del bloque es 512 bytes, el tamaño de cada registro es 45 bytes, el factor de bloque es 11, los registros son de tamaño fijo, y la organización de registros en bloques es no-expandible, ¿cuántos bloques se necesitarían para almacenar el archivo de datos en disco?
- a) 4.545                                      b) 4.546                                      c) 97                                      d) 98                                      e) ninguno
22. [3 pts] Asumiendo que se tiene un archivo de datos ordenado con 3.000 registros, donde el factor de bloque es 30, los registros son de tamaño fijo, la organización de registros en bloques es no-expandible, ¿cuántos bloques se necesitarían leer, en promedio, para buscar un registro por un campo distinto al de ordenamiento?
- a) 6                                      b) 7                                      c) 50                                      d) 100                                      e) ninguno

23. [3 pts.] Asuma que existe un archivo de datos ordenado con 8.880 registros, donde el tamaño del bloque es 8.192 bytes, el tamaño de cada registro es 256 bytes, el factor de bloque es 32, los registros son de tamaño fijo, la organización de registros en bloques es no-expandible, y la cantidad de bloques del archivo de datos es 278. Además, suponga que existe un índice primario sobre este archivo de datos, cuyos registros tienen un tamaño de 64 bytes. ¿Cuántos bloques se necesitan para almacenar el índice primario en disco?

- a) 2                      b) 3                      c) 69                      d) 70                      e) ninguno

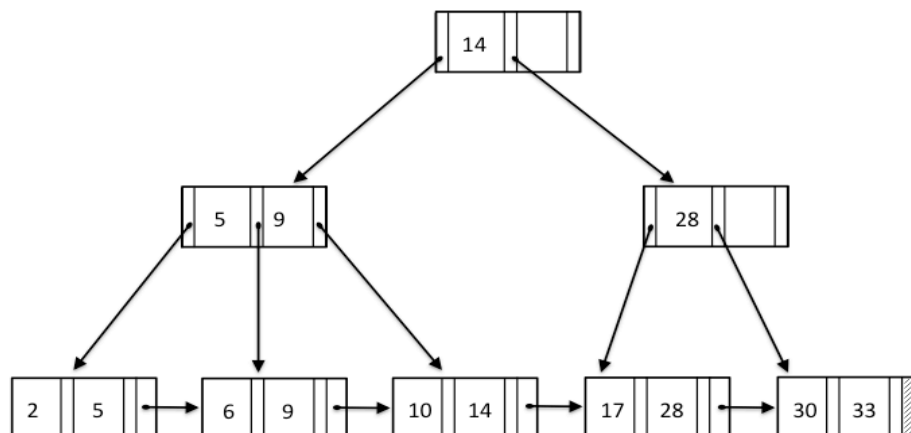
24. [3 pts.] Asumiendo que existe un archivo de datos ordenado con 5.000 registros, donde el tamaño del bloque es 800 bytes, el tamaño de cada registro es 80 bytes, el factor de bloque es 10, los registros son de tamaño fijo, la organización de registros en bloques es no-expandible, y la cantidad de bloques del archivo de datos es 500. Suponga además que existe un índice secundario sobre este archivo de datos, cuyos registros tienen un tamaño de 40 bytes y el factor de bloque para el índice es 20. ¿Cuántos bloques se necesitarían leer para buscar y recuperar un registro por medio del índice secundario?

- a) 8                      b) 9                      c) 10                      d) 125                      e) ninguno

### Sección B [28 pts.]

Las preguntas de esta sección son de desarrollo o de falso/verdadero.

25. [8 pts.] La siguiente figura muestra el contenido de un índice implementado mediante un **árbol B+** de orden  $p=3$ . Muestre la estructura de este árbol después de insertar el registro con valor 4.





26. [20 pts.] Indique si las siguientes afirmaciones son falsas (F) o verdaderas (V).

- a) \_\_\_\_ La unidad mínima de almacenamiento de datos en disco, y de transferencia entre disco y memoria es el bloque.
- b) \_\_\_\_ El tiempo que tarda la cabeza de lectura del disco en posicionarse sobre la pista donde está el dato buscado se llama tiempo de latencia.
- c) \_\_\_\_ La organización de archivos más eficiente para insertar registros y menos eficiente para buscar es el *hash*.
- d) \_\_\_\_ Un árbol B+ garantiza un orden de duración  $O(\log_p n)$  para todas sus operaciones.
- e) \_\_\_\_ Es posible usar búsqueda binaria únicamente sobre archivos de índices.
- f) \_\_\_\_ El índice secundario es el único índice que no requiere que el archivo de datos esté físicamente ordenado por el campo de indexación.
- g) \_\_\_\_ Es posible que un archivo de datos posea múltiples índices primarios.
- h) \_\_\_\_ Un índice raro es aquel que tiene una entrada en el índice por cada registro del archivo de datos.
- i) \_\_\_\_ Un índice agrupado y uno primario no pueden coexistir.
- j) \_\_\_\_ Los cursores resuelven el problema de desajuste de impedancia.