



Grado en Ingeniería Informática Examen teórico – Convocatoria de Junio de 2013

Nombre: _____

Primera parte

1. (3 puntos) Sean las relaciones R y S con los siguientes parámetros:

R(a,b,c)	S(c,d)
N(R) = 5000	N(S)=200
V(R,a)=5000	
V(R,b)=3000	
V(R,c)=5	V(S,c)=5
	V(S,d)=40
Size(a)=20	
Size(b)=60	
Size(c)=20	Size(c)=20
	Size(d)=40

Propón dos planes lógicos para la siguiente consulta e indica qué plan físico se elegirá en base al número de bloques que se utilizan, teniendo en cuenta que el tamaño de bloque es de 2KB, que la cabecera es de 20B y que en memoria sólo cabe un bloque:

$$\Pi_{a,d}(\sigma_{b=b_1}(R \text{ JOIN } S))$$

Pista: uno de los planes es igual que el de la consulta; el otro se obtiene moviendo la selección.

2. Enuncia brevemente usando tus propias palabras (o con la fórmula si la recuerdas y lo prefieres):
- (1 puntos) ¿De qué depende el tiempo de búsqueda de un registro por su clave en un fichero ASI (archivo secuencial indexado) con índice denso y zona de desbordamiento, si la búsqueda se realiza por el valor de clave?
 - (1 puntos) ¿Cómo se calcula la inserción en un fichero AAD (archivo de acceso directo) con encadenamiento abierto?
 - (1 puntos) ¿Qué información se puede consultar a través de las vistas de catálogo con prefijo USER_? ¿Y con las vistas con prefijo ALL_? ¿Y con las vistas con prefijo DBA_?

Segunda parte

3. Considera el siguiente par de transacciones entrelazadas (y que no se consideran bloqueos)
- (0,5 puntos) Rellena la tabla de modificaciones en los espacios vacíos de la derecha considerando que los valores iniciales son A=2 y B=3
 - (1 puntos) Si el sistema sufre un fallo justo después de la sentencia "Escribe (C,t)", ¿qué hará el gestor de recuperaciones con la transacción 1? ¿Y con la transacción 2? ¿Lo que hace con la transacción 2 depende de algún otro factor que no aparece en la transacción pero puede estar escrito en la tabla de modificaciones en el momento del fallo?



T1	T2	T _i	Estado	Operación	Dato	V antiguo	V nuevo
Lee (A, x)							
x := x + 1							
Escribe (A, x)	Lee (A, y)						
	Lee (B, z)						
	t := y + z						
	Escribe (B, t)						
	Escribe (C, t)						

4. Considera las siguientes transacciones y responde a las siguientes cuestiones:

T1	T2	T3
Lee (A, x)		
x := x + 3		
	Lee (A, t)	Lee (C, u)
	t := t * 2	Lee (A, v)
	Escribe (A, t)	w := u + v
Escribe (A, x)		Escribe (C, w)
Lee (C, z)		
z := z + x		
Escribe (C, z)		

- (0,5 puntos)** dibuja el grafo de precedencia indicando qué transacción espera, a qué transacción espera, por culpa de qué átomo tiene que esperar y a causa de qué modo de acceso al átomo.
- (1 puntos)** ¿en qué orden se ejecutan las transacciones según el **algoritmo de ordenación parcial**?
- (1 puntos)** ¿en qué orden se ejecutan las transacciones según el **método de bloqueo en dos fases** suponiendo que: el bloqueo del átomo se realiza justo antes de la primera lectura de dicho átomo, que los desbloques se

producen después de la última sentencia de la transacción y considerando que las *lecturas* son *no protegidas* (M1) y las *escrituras protegidas* (M4)? ¿Existe situación de interbloqueo (*deadlock*)? ¿En caso de existir, cómo se resuelve?

Pista: M1 es compatible con M4, M1 es compatible con M1 pero M4 no es compatible con M4

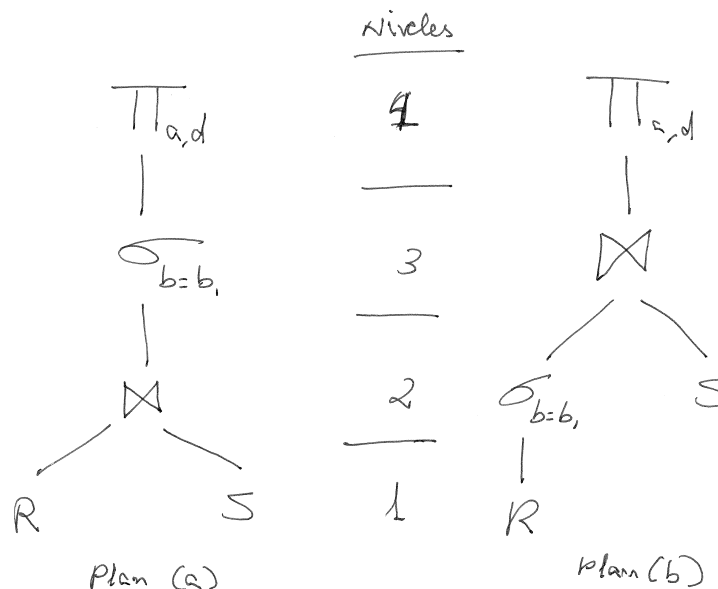
Notas:

- Duración del examen:** 1 hora y media
- Todos los alumnos y todas las alumnas tienen que realizar los ejercicios de la segunda parte.
- Aquellos alumnos y aquellas alumnas que hayan superado la prueba parcial, no están obligados a realizar los ejercicios de la primera parte. En ese caso, la calificación obtenida en dicha prueba se escalará sobre 6.

Resolución

Pregunta 1

Atendiendo al enunciado, los dos planes lógicos propuestos son:



Teniendo en cuenta los dos planes propuestos, calculemos ahora el número de bloques empleados en cada uno de los cuatro niveles para ver qué plan físico sería el escogido para ejecutar.

Plan (a)

En el plan (a), tenemos dos relaciones R y S que tienen que ser leídas:

$$L(R) = 20 \text{ B} + 60 \text{ B} + 20 \text{ B} = 100 \text{ B}$$

$$\text{Bfr}(R) = \text{parte entera}((B - C) / L(R)) = \text{parte entera}((2048 \text{ B} - 20 \text{ B}) / 100 \text{ B}) = \text{parte entera}(20,28) = 20$$

$$\text{Nbloques}(R) = \text{redondeo hacia arriba}(N(R) / \text{Bfr}(R)) = \text{redondeo hacia arriba}(5000 / 20) = 250 \text{ bloques}$$

Teniendo en cuenta que la clave de R sólo puede ser a puesto que es el único atributo que tiene tantos valores como tuplas, y que queremos reunir con la relación S que debe estar ordenada por c (que es el atributo de reunión), es necesario reordenar la relación R para poder reunirla con la relación S .

La reordenación de R consiste en reordenar los bloques de R internamente y entre sí, por lo que el algoritmo más eficiente de ordenación que conocemos tarda un tiempo aproximado de $250 \cdot \log_2(250) = 250 \cdot 7,965784285 = 1991,446071166$, es decir, aproximadamente requiere la transferencia de 1992 bloques para reordenar los registros de R .

Antes de seguir, necesitamos saber el número de bloques de S :

$$L(S) = 20 \text{ B} + 40 \text{ B} = 60 \text{ B}$$

$$\text{Bfr}(S) = \text{parte entera}((B - C) / L(S)) = \text{parte entera}((2048 \text{ B} - 20 \text{ B}) / 60 \text{ B}) = \text{parte entera}(33,8) = 33$$

$N_{\text{bloques}}(S) = \text{redondeo hacia arriba } (N(S) / B_{\text{fr}}(S)) = \text{redondeo hacia arriba } (200 / 33) = \text{redondeo hacia arriba } (6,060606061) = 7 \text{ bloques}$

Para conseguir la reunión natural, sólo necesitamos recorrer las dos relaciones ordenadas por c :

250 bloques + 7 bloques

y escribir la relación resultante:

$L(\text{JOIN}) = 20 B + 60 B + 20 B + 40 B = 140 B$

$B_{\text{fr}}(\text{JOIN}) = \text{parte entera } ((B - C) / L(\text{JOIN})) = \text{parte entera } ((2048 B - 20 B) / 140 B) = \text{parte entera } (14,485714286) = 14$

$N(\text{JOIN}) = N(R) \cdot N(S) / \max\{V(R,c), V(S,b)\} = 5000 \cdot 200 / 5 = 200000$

$N_{\text{bloques}}(\text{JOIN}) = \text{redondeo hacia arriba } (N(S) / B_{\text{fr}}(\text{JOIN})) = \text{redondeo hacia arriba } (200000 / 14) = \text{redondeo hacia arriba } (14285,714285714) = 14286 \text{ bloques}$

Cuando ya tenemos la reunión, es necesario aplicar la selección, es decir, leer los bloques de la reunión, aplicarla y escribir el resultado de la selección:

$N(\text{SELECCIÓN}) = 200000 / 3000 = 66,666666667$, o sea, aproximadamente 67 tuplas

$N_{\text{bloques}}(\text{SELECCIÓN}) = \text{redondeo hacia arriba } (N(\text{SELECCIÓN}) / B_{\text{fr}}(\text{JOIN})) = \text{redondeo hacia arriba } (67 / 14) = \text{redondeo hacia arriba } (4,785714286) = 5 \text{ bloques}$

Por último, habría que aplicar la proyección a estos cinco bloques:

$L(\text{PROYECCIÓN}) = 20 B + 40 B = 60 B$

$B_{\text{fr}}(\text{PROYECCIÓN}) = \text{parte entera } ((B - C) / L(\text{PROYECCIÓN})) = \text{parte entera } ((2048 B - 20B) / 60B) = \text{parte entera } (33,8) = 33$

$N_{\text{bloques}}(\text{PROYECCIÓN}) = \text{redondeo hacia arriba } (N(\text{PROYECCIÓN}) / B_{\text{fr}}(\text{PROYECCIÓN})) = \text{redondeo hacia arriba } (67 / 33) = \text{redondeo hacia arriba } (2,03030303) = 3 \text{ bloques}$

Teniendo en cuenta que el resultado de cada operación intermedia tiene que ser escrito a disco y leído después, podemos resumir que el número de bloques leídos y escritos en este plan físico es de:

- Reordenar R requiere transferir 1992 bloques
- Mezclar R reordenado y S (resolver el JOIN) requiere leer 250 + 7 bloques y escribir 14286 bloques
- Aplicar la selección requiere leer 14286 bloques y escribir 5 bloques
- Aplicar la proyección requiere leer 5 bloques y escribir 3 bloques

En total, para resolver este plan es necesario transferir 30834 bloques

Plan (b)

En este plan, podemos aprovecharnos de algunos cálculos realizados para el plan anterior.

Para aplicar la selección sobre R , necesitamos leer los bloques de dicha relación:

$N_{\text{bloques}}(R) = 250 \text{ bloques}$

Ahora, aplicamos la selección y calculamos el número de bloques:

$N(\text{SELECCIÓN}) = N(R) / V(R,b) = 5000 / 3000 = 1,666666667$, es decir, aproximadamente 2 registros

$N_{\text{bloques}}(\text{SELECCIÓN}) = \text{redondeo hacia arriba } (N(\text{SELECCIÓN}) / B_{\text{fr}}(R)) = \text{redondeo hacia arriba } (2 / 20) = \text{redondeo hacia arriba } (0,1) = 1 \text{ bloque}$

Para realizar la reunión, es preciso que el resultado de la selección esté ordenado, pero para ordenar

un bloque sólo se requiere el propio bloque, luego la reordenación requiere una transferencia de bloque.

Ahora que la selección de R está ordenada por c , ya podemos mezclar la selección de R con S , lo que requiere leer 1 bloque y 7 bloques más, y el resultado de la reunión tendrá:

$$N(\text{JOIN}) = N(\text{SELECCIÓN}) \cdot N(S) / \max\{V(R,c), V(S,b)\} = 2 \cdot 200 / 5 = 80 \text{ registros}$$

$$\text{Nbloques}(\text{JOIN}) = \text{redondeo hacia arriba} (N(\text{JOIN}) / \text{Bfr}(\text{JOIN})) = \text{redondeo hacia arriba} (80 / 14) = \text{redondeo hacia arriba} (5,714285714) = 6 \text{ bloques}$$

Sobre el resultado de la reunión, es necesario aplicar la proyección:

$$\text{Nbloques}(\text{PROYECCIÓN}) = \text{redondeo hacia arriba} (N(\text{JOIN}) / \text{Bfr}(\text{PROYECCIÓN})) = \text{redondeo hacia arriba} (80 / 33) = \text{redondeo hacia arriba} (2,424242424) = 3 \text{ bloques}$$

Teniendo en cuenta que el resultado de cada operación intermedia tiene que ser escrito a disco y leído después, podemos resumir que el número de bloques leídos y escritos en este plan físico es de:

- para aplicar la selección sobre R , necesitamos leer 250 bloques y escribir 1 bloque
- para reordenar el resultado de la selección necesitamos leer 1 bloque y escribir 1 bloque
- para aplicar la reunión natural, necesitamos leer $7 + 1$ bloques y escribir 6 bloques
- para aplicar la proyección, necesitamos leer 6 bloques y escribir 3 bloques

En total, para resolver este plan es necesario transferir 276 bloques

Como conclusión, entendemos que el plan físico seleccionado como mejor es el **plan (b)** puesto que requiere transferir 276 bloques, contra los 30834 bloques que requiere el plan (a).

Ejercicio 2

- a) Dependerá del tiempo que se tarda en aplicar la búsqueda binaria sobre el índice más el tiempo que se tarda en recorrer la cadena de desbordamiento si no se encuentra en el fichero maestro.
- b) El tiempo que se tarda en calcular la posición que le corresponde más el tiempo de escribir el registro o, si hay colisión, de escribir el registro en la zona de desbordamiento y modificar el registro que produce la colisión para apuntar a este nuevo.
- c) Las vistas de catálogo USER_ contienen información sobre todos los objetos creados por el usuario que la consulta, las vistas de catálogo ALL_ contienen información sobre todos los objetos a los que el usuario tiene acceso y las vistas de catálogo DBA_ contienen información sobre todos los objetos del sistema.

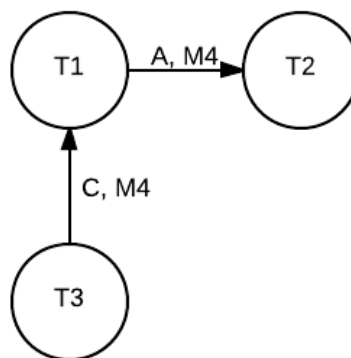
Ejercicio 3

T1	T2	T _i	Estado	Operación	Dato	V antiguo	V nuevo
Lee (A, x)		1	activa				
x := x + 1							
Escribe (A, x)	Lee (A, y)	2	activa				
	Lee (B, z)	1		update	A	2	3
	t := y + z	1	commit				
	Escribe (B, t)	2		update	B	3	5
	Escribe (C, t)	2		insert	C		5
		2	commit				

- b) Cuando se produce el fallo, la transacción 1 hace tiempo que terminó por lo que el sistema de recuperación ejecutaría una orden *REDO* sobre la transacción 1. Sin embargo, con la transacción 2 pueden ocurrir dos cosas:
1. si el fallo se produce después de haber escrito el registro *COMMIT*, el sistema de recuperación ejecutaría una orden *REDO* sobre la transacción 2, o
 2. si el fallo se produce antes de haber escrito el registro *COMMIT*, el sistema de recuperación ejecutaría una orden *UNDO* sobre la transacción 2.

Ejercicio 4

a)



- b) Las operaciones que se ven afectadas según el algoritmo de operación parcial y en el orden en el que intentan ejecutarse serían:
- Lee (T1, A), Lee (T2, A), Lee (T3, C), Lee (T3, A), Escribe (T2, A), Escribe (T1, A), Escribe (T3, C), Lee (T1, C), Escribe (T1, C)

Cuando se ejecuta *Lee (T2, A)*, no ha escrito ninguna transacción, luego se puede ejecutar sin problemas. *Lee (T3, C)* se puede ejecutar sin problemas, dado que es la primera lectura sobre C. *Lee (T3, A)* no ha escrito ninguna transacción, luego se puede ejecutar sin problemas. La operación *Escribe (T2, A)* falla puesto que la última lectura ha sido realizada por una transacción más joven (T3) y es necesario abortar la transacción T2.

Lee (T1, A), Lee (T3, C), Lee (T3, A), Escribe (T1, A), Escribe (T3,C), Lee (T1, C), Escribe (T1,C), Lee (T4, A), Escribe (T4, A)

La operación *Escribe (T1, A)* falla puesto que la última lectura ha sido realizada por una transacción más joven (T3) y es necesario abortar la transacción T1.

Lee (T3, C), Lee (T3, A), Escribe (T3,C), Lee (T4, A), Escribe (T4, A), Lee (T5, A), Escribe (T5, A), Lee (T5, C), Escribe (T5,C)

Escribe (T3, C) funciona puesto que el último que ha leído es T3 y nadie ha escrito antes.

Lee (T4, A) funciona puesto que el último que ha escrito es T3.

Escribe (T4, A) funciona puesto que el último que ha leído es T4 y el último que ha escrito es T3.

Lee (T5, A) funciona puesto que el último que ha escrito es T4.

Escribe (T5, A) funciona puesto que el último que ha leído es T5 y el último que escribió es T4.

Lee (T5, C) funciona puesto que el último que escribió fue T3.

Escribe (T5, C) funciona puesto que el último que leyó fue T5 y el último que escribió fue T3.

En resumen, las transacciones se ejecutan en el siguiente orden: **T3, T2, T1**

- c) Tomando en consideración el momento de los bloqueos y los desbloqueos, las transacciones deberían reescribirse como se ve en la tabla a continuación.

Cuando T2 intenta bloquear A en el modo M4, T1 ya lo ha hecho, y el modo es incompatible consigo mismo, de modo que T2 tendrá que esperar a que T1 termine con A.

Cuando T3 llega a bloquear C en modo M4, es la primera, por lo que consigue bloquearlo. Después de eso, intenta bloquear A en modo M1, que es compatible con el modo M4 (en el que T1 ha bloqueado, y por el que T2 está esperando en la lista), de modo que lo bloquea si sigue adelante. Hasta que completa todas sus operaciones, incluidos los desbloqueos de C y A.

Cuando T1 necesita bloquear C, T3 ya ha desbloqueado, por lo que puede bloquear sin problemas. Y continúa su ejecución hasta el desbloqueo de A y C, momento en el cual, T2 puede reanudarse, bloquear A y realizar su ejecución.

T1	T2	T3
<i>Lock (A, M4)</i>		
Lee (A, x)		
$x := x + 3$		
	<i>Lock (A, M4)</i>	<i>Lock (C, M4)</i>
	Lee (A, t)	Lee (C, u)
		<i>Lock (A, M1)</i>
	$t := t * 2$	Lee (A, v)
	Escribe (A, t)	$w := u + v$
	<i>Unlock (A)</i>	
Escribe (A, x)		Escribe (C, w)
		<i>Unlock (C)</i>
		<i>Unlock (A)</i>
<i>Lock (C, M4)</i>		
Lee (C, z)		
$z := z + x$		
Escribe (C, z)		
<i>Unlock (A)</i>		
<i>Unlock (C)</i>		

De este modo, el orden de ejecución (por el momento en que terminan) de las transacciones será **T3, T1, T2**.