

Corrección Examen de contenidos teóricos (Junio de 2014)

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)=300	
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 el plan lógico más eficiente para la siguiente consulta y justifica numéricamente por qué es más eficiente en base al número de bloques que emplea cada uno y 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:

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

El plan más eficiente para la consulta planteada sería el generado por el árbol de expresión correspondiente a la consulta:

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

Teniendo en cuenta que cada operación de E/S representa la transferencia de un bloque de datos entre el disco y la memoria, o viceversa, tendremos que calcular el número de bloques transferidos en total.

A fin de justificar que el plan que propongo es más eficiente que el planteado en el enunciado, tendré que calcular el número de bloques de E/S necesarios para uno y otro plan.

Comenzando por el planteado en el enunciado:

El número de bloques que almacenan la relación R es de:

$$B(R) = \text{redondeo hacia arriba} \left(\frac{N(R)}{Bfr(R)} \right)$$

Para ello, hay que calcular el factor de bloqueo de R como:

$$Bfr(R) = \text{parte entera} \left(\frac{B-C}{L(R)} \right)$$

que dependerá de la longitud del registro de R:

$$L(R) = 20B + 60B + 20B = 100B$$

De modo que:

$$Bfr(R) = \text{parte entera} \left(\frac{2048-20}{100} \right) = 20$$

$$B(R) = \text{redondeo hacia arriba} \left(\frac{5000}{20} \right) = 250 \text{ bloques}$$

Si aplicamos los mismos cálculos para el número de bloques de la relación S:

$$L(S) = 20B + 40B = 60B$$

$$Bfr(S) = \text{parte entera} \left(\frac{2048 - 20}{60} \right) = 33$$

$$B(S) = \text{redondeo hacia arriba} \left(\frac{200}{33} \right) = 7 \text{ bloques}$$

Como ya se sabe, hay varios algoritmos para resolver la operación de reunión natural, pero la más eficiente, consiste en ordenar por el atributo de reunión aquella relación que no lo está (sólo una de las dos). Entendemos que R no lo está por lo que tenemos que aplicar un algoritmo de ordenación que requiere realizar un número de lecturas de:

$$B(R) \cdot \log_2(B(R)) = 250 \cdot \log_2(250) = 1991,45 \approx 1992 \text{ bloques}$$

Ahora que las dos relaciones están ordenadas por el atributo c, podemos realizar la operación de reunión natural entre las dos. Habrá que leer:

$$250 + 7 = 257 \text{ bloques}$$

y escribir los bloques resultantes de la operación de reunión:

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

$$Bfr(\text{JOIN}) = \text{parte entera} \left(\frac{B - C}{L(\text{JOIN})} \right) = \text{parte entera} \left(\frac{2048 - 20}{140} \right) = 14$$

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(\text{JOIN})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{14} \right)$$

El número de registros de la operación de reunión natural viene dado por:

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

por lo que el número de bloques resultantes que tendremos que escribir en disco será de:

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{200000}{14} \right) = 14286 \text{ bloques}$$

A fin de aplicar la operación de selección sobre el atributo d, tendremos que leer de nuevo:

$$14286 \text{ bloques}$$

y el número de bloques que se transferirán al disco como resultado de la selección será de:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{SELEC})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{JOIN})} \right)$$

La estimación del número de tuplas que cumplen una condición de desigualdad de mayor o menor es de un tercio del total de las tuplas de la relación sobre la que se aplica la operación de selección.

$$N(\text{SELEC}) = \frac{N(\text{JOIN})}{3} = \frac{200000}{3} \approx 66667$$

por lo que:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{66667}{14} \right) = 4762 \text{ bloques}$$

Por último, para realizar la operación de proyección, tendremos que leer:

4762 bloques

y escribir el número de bloques resultantes de dicha operación de proyección:

$$B(\text{PROY}) = \text{redondeo hacia arriba} \left(\frac{N(\text{PROY})}{Bfr(\text{PROY})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY})} \right)$$

$$Bfr(\text{PROY}) = \text{parte entera} \left(\frac{B-C}{L(\text{PROY})} \right) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY})} \right)$$

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

Con lo que:

$$Bfr(\text{PROY}) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY})} \right) = \text{parte entera} \left(\frac{2048-20}{60} \right) = 33$$

$$B(\text{PROY}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY})} \right) = \text{redondeo hacia arriba} \left(\frac{66667}{33} \right) = 2021 \text{ bloques}$$

De modo que el número total de lecturas y escrituras de bloques necesarios para resolver este plan físico es de:

$$1992 + 250 + 7 + 14286 \cdot 2 + 4762 \cdot 2 + 2021 = 42366 \text{ bloques}$$

El número de bloques involucrado en el plan físico de la consulta que propongo:

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

sería el que se muestra a continuación.

Como se ha mostrado en el plan físico de la consulta anterior, el número de bloques de la relación R es de 250 bloques, que habrá que leer para aplicar la operación de selección sobre la misma. El número de bloques resultante de dicha operación viene dado por los siguientes cálculos:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{SELEC})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(R)} \right)$$

$$N(\text{SELEC}) = \frac{N(R)}{3} = \frac{5000}{3} \approx 1667$$

con lo que:

$$B(\text{SELEC}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(R)} \right) = \text{redondeo hacia arriba} \left(\frac{1667}{20} \right) = 84 \text{ bloques}$$

Sobre el resultado de la operación de selección, es necesario aplicar la primera operación de proyección sobre los atributos a y c, que arroja un número de bloques resultantes como sigue:

$$B(\text{PROY1}) = \text{redondeo hacia arriba} \left(\frac{N(\text{PROY1})}{Bfr(\text{PROY1})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY1})} \right)$$

$$Bfr(\text{PROY1}) = \text{parte entera} \left(\frac{B-C}{L(\text{PROY1})} \right) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY1})} \right)$$

$$L(\text{PROY1}) = 20B + 20B = 40B$$

Con lo que:

$$Bfr(\text{PROY1}) = \text{parte entera} \left(\frac{2048-20}{L(\text{PROY1})} \right) = \text{parte entera} \left(\frac{2048-20}{40} \right) = 50$$

$$B(\text{PROY1}) = \text{redondeo hacia arriba} \left(\frac{N(\text{SELEC})}{Bfr(\text{PROY1})} \right) = \text{redondeo hacia arriba} \left(\frac{1667}{50} \right) = 34 \text{ bloques}$$

El primer paso de la reunión natural, consiste en re-ordenar la relación R por el atributo de reunión c , lo cual tiene un coste en bloques de:

$$B(\text{PROY1}) \cdot \log_2(\text{PROY1}) = 34 \cdot \log_2(34) \approx 173 \text{ bloques}$$

Con respecto a la relación S , está almacenada en 7 bloques (como se ha visto en el plan físico de la consulta planteada por el enunciado) que habrá que leer, junto con los 34 bloques de la proyección calculada. La reunión natural arrojará un resultado en bloques de:

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(\text{JOIN})} \right)$$

$$N(\text{JOIN}) = \frac{N(\text{PROY1}) \cdot N(S)}{\max\{V(R, c), V(S, c)\}} = \frac{N(\text{PROY1}) \cdot 200}{\max\{5, 5\}} = \frac{N(\text{SELEC}) \cdot 200}{5} = \frac{1667 \cdot 200}{5} = 66680$$

$$Bfr(\text{JOIN}) = \text{parte entera} \left(\frac{B - C}{L(\text{JOIN})} \right) = \text{parte entera} \left(\frac{2048 - 20}{L(\text{JOIN})} \right)$$

$$L(\text{JOIN}) = 20B + 20B + 40B = 80B$$

con lo que queda:

$$Bfr(\text{JOIN}) = \text{parte entera} \left(\frac{2028}{80} \right) = 25$$

$$B(\text{JOIN}) = \text{redondeo hacia arriba} \left(\frac{66680}{25} \right) = 2668 \text{ bloques}$$

Por último, y sobre el resultado de la reunión natural, será necesario leer los 2668 bloques para aplicar la segunda operación de proyección para eliminar el atributo c , lo que nos deja con un número de bloques como sigue:

$$B(\text{PROY2}) = \text{redondeo hacia arriba} \left(\frac{N(\text{PROY2})}{Bfr(\text{PROY2})} \right) = \text{redondeo hacia arriba} \left(\frac{N(\text{JOIN})}{Bfr(\text{PROY2})} \right)$$

$$Bfr(\text{PROY2}) = \text{parte entera} \left(\frac{2048 - 20}{L(\text{PROY2})} \right)$$

$$L(\text{PROY2}) = 20B + 40B = 60B$$

con lo que queda:

$$Bfr(\text{PROY2}) = \text{parte entera} \left(\frac{2048 - 20}{60} \right) = 33$$

$$B(\text{PROY2}) = \text{redondeo hacia arriba} \left(\frac{66680}{33} \right) = 2021 \text{ bloques}$$

De modo que el número total de lecturas y escrituras de bloques necesarios para resolver este plan físico es de:

$$250 + 84 \cdot 2 + 34 + 173 + 34 + 7 + 2668 \cdot 2 + 2021 = 8023 \text{ bloques}$$

opción que supone menos de una quinta parte de las lecturas y escrituras de bloques implicadas por el plan físico correspondiente a la consulta planteada por el enunciado.

2. Enuncia brevemente usando tus propias palabras (o con la fórmula si la recuerdas y lo prefieres):
- a) **(1 punto)** ¿De qué depende el tiempo de búsqueda de un registro por su clave en un fichero ASI (archivo secuencial indexado) multinivel con zona de desbordamiento, si la búsqueda se realiza por el valor de clave?.
 - b) **(1 punto)** Enumera los elementos de los que se compone el nivel interno de Oracle Database®.
 - c) **(1 punto)** Los índices son útiles para búsquedas por valor de clave pero son costosos de mantener. Enumera las cuatro situaciones en las que es conveniente eliminarlos.
-

Apartado (a)

Depende del tiempo necesario para encontrar la clave en el índice raíz, el tiempo necesario para encontrar la clave en cada uno de los índices intermedios y el tiempo para acceder a un registro del fichero maestro por posición, además del tiempo para encontrar el registro en la cadena de desbordamiento, en caso de que no estuviera presente en la posición indicada del fichero maestro.

$$T_F = T_M + (m - 1) \cdot T_{F_i} + T'_F + T_{F \text{ Cadena}}$$

Apartado (b)

La base de datos se compone de varios *tablespaces*, que se componen, a su vez, de varios segmentos, que se componen de varias extensiones, que se componen de varios bloques.

Apartado (c)

Los índices deben eliminarse cuando concurra una de las siguientes situaciones:

- Ya no sirven
- No mejoran la eficiencia
- Hay que cambiar los campos que se indexan
- Hay que rehacerlo

Segunda parte

3. Considera el siguiente trío de transacciones entrelazadas:

- a) **(0,5 puntos)** Rellena la tabla de modificaciones en los espacios vacíos de la derecha considerando que los valores iniciales son S=60, A=61, B=20, D=40 y E=50. Ten en cuenta que las transacciones comienzan antes de la primera lectura y que terminan después de la última sentencia.
- b) **(1 punto)** Si el sistema sufre un fallo justo después de la sentencia “Escribe (D,t)”, ¿qué hará el gestor de recuperaciones con las transacciones? ¿Depende de algún factor lo que hace con la transacción 3? Y si se hubiera realizado un *savepoint* justo antes del fallo, ¿qué haría en ese caso con cada una de las transacciones?

T1	T2	T3	T _i	Estado	Oper.	Dato	V antiguo	V nuevo
Lee (S, x)								
x := x + 1								
Escribe (S, x)	Lee (A, y)							
	y := y + 1	Lee (B, z)						
	Escribe (A, y)	t := z * 2 + 1						
		z := z + 1						
		Escribe (B, z)						
		Escribe (D, t)						
	y := y - 10							
	Escribe (E, y)							

Apartado (a)

T1	T2	T3	T _i	Estado	Oper.	Dato	V antiguo	V nuevo
Lee (S, x)			T1	Start	-	-	-	-
x := x + 1			T1	-	Update	S	60	61
Escribe (S, x)	Lee (A, y)		T2	Start	-	-	-	-
	y := y + 1	Lee (B, z)	T1	Commit	-	-	-	-
	Escribe (A, y)	t := z * 2 + 1	T3	Start	-	-	-	-
		z := z + 1	T2	-	Update	A	61	62
		Escribe (B, z)	T3	-	Update	B	20	21
		Escribe (D, t)	T3	-	Update	D	40	41
	y := y - 10		T3	Commit	-	-	-	-
	Escribe (E, y)		T2	-	Update	E	50	52
			T2	Commit	-	-	-	-

Apartado (b)

Si ocurriera un fallo tras la sentencia “Escribe (D, t)”, el gestor de recuperaciones reharía (*redo*) la transacción 1, cuyo *Commit* encuentra en la tabla. Desharía la transacción 2 (*undo*), puesto que no puede encontrar el *Commit* de dicha transacción. Sin embargo, el enunciado no especifica si el fallo ocurre antes o después de que la transacción 3 haya logrado realizar el *Commit* o no. Si lo ha logrado (está en la tabla) y la transacción se rehace (*redo*) y si no, la transacción se deshace (*undo*). De ese modo, el rehacer o deshacer la transacción 3 depende de que se haya escrito el *Commit* o no.

Si se ha realizado un *Savepoint* justo antes del fallo, no se hace nada con la transacción 1, ya que el *checkpoint* garantizó su escritura a disco. La transacción 2 se deshace (*undo*) y tenemos dos posibles situaciones, de nuevo, con la transacción 3:

- si hay *Commit* para la transacción, no se hace nada (por el mismo motivo por el que no se hace nada con la transacción 1)
- si no hay *Commit*, se deshace (*undo*) al igual que si no hubiera *Savepoint*.

4. Considera la siguiente ejecución y responde a las cuestiones:

Lee (T1, A), Lee (T3, A), Lee (T2, A), Escribe (T3, C), Escribe (T2, B), Escribe (T1, B), Escribe (T2, A), Escribe (T1, A)

- a) **(1 punto)** ¿en qué orden se ejecutan las transacciones según el **algoritmo de ordenación parcial modificado**?
- b) **(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.
- c) **(1 punto)** ¿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

Apartado (a)

El algoritmo de control por marcas de tiempo parcial modificado, utiliza dos controladores por cada átomo A, B y C, uno para la lectura más reciente y otro para la escritura más reciente:

RR(A)	0								
WR(A)	0								

Las operaciones de lectura Lee (T1, A), Lee (T3, A) dejan los controladores como siguen:

RR(A)	0	1	3						
WR(A)	0								

La realización de la operación de lectura Lee (T2, A) no afecta a los controladores, aunque la lectura se realiza con éxito (las lecturas no pueden hacer que una transacción aborte).

La operación de escritura Escribe (T3, C) deja los controladores del átomo C como siguen:

RR(C)	0								
WR(C)	0	3							

La operación de escritura Escribe (T2, B) deja los controladores del átomo B como siguen:

RR(B)	0								
WR(B)	0	2							

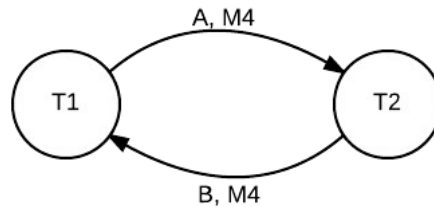
La operación de escritura Escribe (T1, B) deja los controladores del átomo B como estaban, aunque no provocan que la transacción 1 se aborte, debido a que se trata de una escritura de una transacción más antigua después de una escritura de una transacción más nueva. Simplemente, se ignora la escritura pero no se cambian los controladores.

La operación de escritura Escribe (T2, A) obliga a abortar la transacción 2 debido a que una transacción más reciente ha realizado una lectura, en concreto, la transacción 3. Al abortar la transacción 2, todas sus sentencias se agrupan al final del plan de ejecución como transacción 4, y se ejecutan en serie.

La operación de escritura *Escribe* ($T1, A$) obliga a abortar la transacción 1 por la misma razón que se hizo con la transacción 2. Al abortar la transacción 1, todas sus sentencias se agrupan al final del plan de ejecución como transacción 5, y se ejecutan en serie.

De este modo, el plan de ejecución coincide con la serialización de las transacciones $T3, T2$ y $T1$.

Apartado (b)



Apartado (c)

Completamos el plan con las operaciones de bloqueo, siguiendo las condiciones establecidas por el enunciado (transacciones de dos fases con bloqueo justo antes del primer acceso en el modo necesario para cubrir todos los accesos al átomo en esa transacción y con desbloqueo a la finalización de la transacción):

Bloqueo ($T1, A, M4$), *Lee* ($T1, A$), *Bloqueo* ($T3, A, M1$), *Lee* ($T3, A$), *Bloqueo* ($T2, A, M4$), *Lee* ($T2, A$), *Bloqueo* ($T3, C, M4$), *Escribe* ($T3, C$), *Desbloqueo* ($T3, A$), *Desbloqueo* ($T3, C$), *Bloqueo* ($T2, B, M4$), *Escribe* ($T2, B$), *Bloqueo* ($T1, B, M4$), *Escribe* ($T1, B$), *Escribe* ($T2, A$), *Desbloqueo* ($T2, B$), *Desbloqueo* ($T2, A$), *Escribe* ($T1, A$), *Desbloqueo* ($T1, B$), *Desbloqueo* ($T1, A$)

Hay que recordar que el algoritmo emplea para cada átomo un vector de modos compatibles en los que el átomo está siendo usado en ese momento, y una cola de espera que establece qué transacción espera y qué modo está esperando para el átomo.

Podemos obviar el uso del vector puesto que los dos modos incluidos en este enunciado son el $M1$ y el $M4$, y conocemos su compatibilidad por dicho enunciado.

El primer bloqueo *Bloqueo* ($T1, A, M4$) provoca que el vector de estado de A incluya a $M4$, por lo que la transacción 1 obtiene el bloqueo de A .

El bloqueo *Bloqueo* ($T3, A, M1$) no es incompatible con el vector de estado de A ya que $M1$ y $M4$ son compatibles entre sí, por lo que el vector de estado de A incluye a $M1$ y $M4$, y la transacción 3 obtiene el bloqueo de A .

El bloqueo *Bloqueo* ($T2, A, M4$) no es compatible con el vector de estado de A ya que $M4$ es incompatible con el bloqueo $M4$ que ha obtenido la transacción 1, por lo que no se modifica el vector de estado y la cola de espera para el átomo A pasa a contener a la transacción 2 con el modo $M4$: $Q(A) = \{(T2, M4)\}$. La transacción 2 queda a la espera por no haber podido bloquear el átomo A .

El bloqueo *Bloqueo* ($T3, C, M4$) provoca que el vector de estado de C incluya a $M4$, por lo que la transacción 3 obtiene el bloqueo de C .

El desbloqueo *Desbloqueo* ($T3, A$) libera el bloqueo del átomo A y el vector de estado de A se modifica para eliminar este modo, pasando a contener únicamente el modo $M4$. Dado que la cola de espera para A no está vacía, se buscan dentro de ella transacciones compatibles con los modos que quedan activos ($M4$), pero no se encuentra ninguna porque la que queda es la transacción 2 que necesita el modo $M4$, incompatible con $M4$. Por ello, no se desbloquea ninguna transacción.

El desbloqueo *Desbloqueo* ($T3, C$) libera el bloque del átomo C y el vector de estado de C se modifica para eliminar este modo. Dado que era el único modo en uso sobre C , el vector de estado de C queda vacío de nuevo. Puesto que ninguna transacción espera a C en este punto, no se desbloquea ninguna transacción.

Hay que considerar que la transacción 2 sigue bloqueada por culpa de su incapacidad de bloquear el átomo A por lo que no se consideran operaciones de la transacción 2 hasta que no se desbloquee.

El primer bloqueo *Bloqueo* ($T1, B, M4$) provoca que el vector de estado de B incluya a $M4$, por lo que la transacción 1 obtiene el bloqueo de B .

El desbloqueo *Desbloqueo* ($T1, B$) libera el bloqueo del átomo B y el vector de estado de B se modifica para eliminar este modo, pasando a estar vacío. Puesto que ninguna transacción espera a B en este punto (la transacción 2 sigue bloqueada por el átomo A), no se desbloquea ninguna transacción.

El desbloqueo *Desbloqueo* ($T1, A$) libera el bloqueo del átomo A y el vector de estado de A se modifica para eliminar este modo, pasando a estar vacío. Dado que la cola de espera para A no está vacía, se buscan dentro de ella transacciones compatibles con los modos que quedan activos (ningún modo activo), por lo que la transacción 2 pasa a desbloquearse y consigue bloquear el átomo A en modo $M4$.

Puesto que no queda ninguna transacción activa, el resto de la transacción 2 se ejecuta sin detenerse ni bloquearse.

De este modo, el orden de ejecución de las transacciones es $T3, T1$ y $T2$.

No se da situación de interbloqueo en esta ejecución particular, pero el grafo de precedencia avisa de una posible situación de interbloqueo antes de comenzar la ejecución. Dicho aviso, provocaría que no se permitiera ejecutar las dos transacciones a la vez y habría que abortar una nada más comenzar. Para elegir cuál abortar, se podría elegir entre:

- la que bloquea el átomo más solicitado (pero tanto 1 como 2 bloquean A y habría que aplicar otro criterio de elección o elegir aleatoriamente)
- la que bloquea más átomos (pero tanto 1 como 2 bloquean el mismo número de átomos y habría que aplicar otro criterio de elección o elegir aleatoriamente)

Ejercicio extra

5. **(1 punto)** ¿Cómo se realiza el control de bloqueo por protocolo de copia primaria en un sistema gestor de bases de datos distribuido?
-

Se solicita el bloqueo al nodo de la copia primaria. En el caso de la lectura, se realiza la lectura sobre la copia primaria y, en el caso de la escritura, la copia primaria se encarga de sobre-escribir todas las demás.