

# UNIVERSIDAD GALILEO

## Programación VI (Base de Datos)

### Examen Parcial 2

**Primera serie (30 Puntos):** Desarrolle cada una de las siguientes preguntas

1. ¿Cuáles son las propiedades ACID?
2. ¿Qué es un deadlock? De un ejemplo  
Es cuando varias transacciones se quedan esperando a otra transacción y a la vez esa transacción espera de la otra transacción

T(1)	T(2)
X(A)	
	X(B)
Se va a dormir ← X(B)	
	X(A) → Se va a dormir

3. Describa las políticas de robar frames y no forzar  
Esta consiste en que si no existe espacio en memoria se quita una página de las que está allí y se escribe a disco para que se pueda agregar la nueva página
4. Explique los pasos a seguir para la construcción de un índice
5. ¿Qué debe incluir la descripción de la carga de trabajo?  
Las consultas que más se utilizan, actualizaciones que más se hacen y el tiempo en el que estas se ejecutan.
6. ¿En qué consiste la descomposición horizontal?  
En dividir una tabla para que no sea tan pesada con respecto a datos, pero será un poco más pesada ya que para obtener toda la información se necesita hacer una unión
7. ¿En qué consiste la afinación de índices? ¿Por qué es necesaria?
8. ¿Qué es denormalización?  
Dejar la forma normal más pequeña
9. Explique el control de acceso mandatorio  
A cada usuario se le define un acceso sobre cada objeto
10. Explique el control de acceso discrecional  
Se dan permisos al usuario sobre las transacciones

**Segunda serie (70 Puntos):** Resuelva los siguientes problemas

**Problema 1 (15 Puntos)**

Considera las siguientes clases de calendarizaciones:

- Serializable
- Serializable respecto a conflictos
- Recuperables
- Evitan aborts en cascada.

Para cada una de las calendarizaciones mostradas, diga a que clase pertenece y cual no, y si en algún caso no pude determinar si pertenece o no, explíquelo brevemente.

Las acciones están listadas de acuerdo con su orden en la calendarización, teniendo como prefijo el nombre de la transacción.

- a) T1:R(X),  
    T2:R(X),  
    T1:W(X),  
    T2:W(X)
- b) T1:R(X),  
    T2:R(Y),  
        T3:W(X),  
    T2:R(X),  
    T1:R(Y)
- c) T1:R(X),  
    T2:W(X),  
    T1:W(X),  
        T2:ABORT,  
    T1:COMMIT
- d) T1  
    T2:R(X),  
        T3:W(X),  
        T3:COMMIT,  
    T1:W(Y),  
    T1:COMMIT,  
        T2:R(Y),  
        T2:W(Z),  
        T2:COMMIT
- e) T1:R(X),  
    T2:W(X),  
    T1:W(X),  
        T3:R(X),  
    T1:COMMIT,  
        T2:COMMIT,  
        T3:COMMIT

Serializable	Serializable respecto a conflictos	Recuperables	Evita Aborts en cascada
No	No	No	No
Si	Si	No	No
Si	No	Si	No
Si	Si	Si	Si
No	Si	Si	Si

### Problema 2 (15 Puntos)

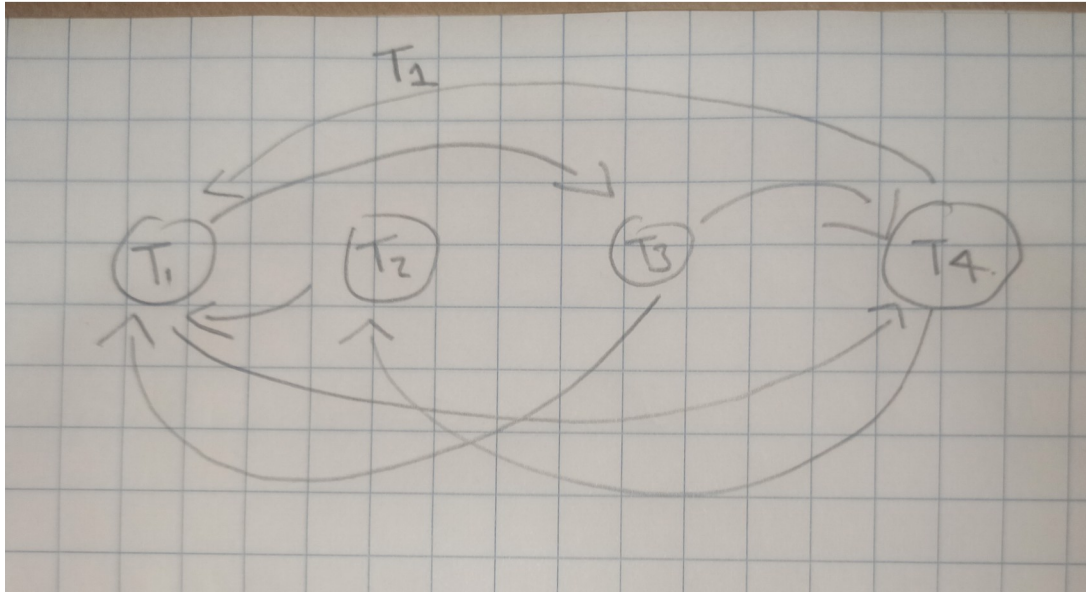
Suponga que un DBMS reconoce adicional a read y write las acciones increment la cual incrementa el valor de un objeto integer en 1 y la acción decrement. Una transacción que incrementa un objeto no necesita saber el valor al igual que decrement por lo que son versiones de un blind write. Además de los bloqueos share y exclusive, dos bloqueos adicionales son soportados: Un objeto debe de estar bloqueado en modo I antes de incrementar y bloqueado en modo D antes de disminuir el valor. Un bloqueo I es compatible con otro bloqueo tipo I o D en el mismo objeto pero no con S y X.

Muestre como el uso de bloqueos I y D pueden incrementar concurrencia (Muestre una calendarización permitida por 2PL estricto que únicamente utilicé bloqueos S y X. Explique como el uso de bloqueos I y D permite que mas acciones sean intercaladas mientras se sigue respetando el protocolo de bloqueo.

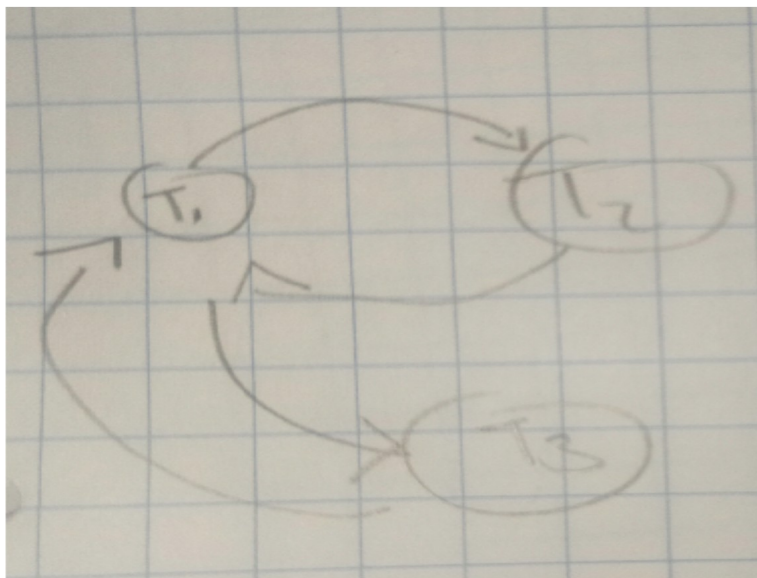
### Problema 3 (15 Puntos)

T	T1	T2	T3	T4
1				R(tax)
2	R(salary)			
3				W(tax)
4		R(tax)		
5		W(tax)		
6	R(tax)			
7	W(salary)			
8			R(salary)	
9	W(tax)			
10			W(salary)	
11				R(salary)
12				W(salary)

- a. Dibuje la grafica de dependencias para esta calendarización



- b. A que serialización equivale esta calendarización. Si no existe por favor indíquelo.  
 c. Asuma que la transacción T4 no se ejecuto para nada. ¿Cuál es la grafica de dependencia para este caso?



- d. A que

serialización equivale esta segunda calendarización. Si no existe por favor indíquelo.

#### Problema 4 (15 Puntos)

Considere una base de datos con la siguiente jerarquía de objetos: Base datos (D) la cual contiene dos archivos (F1 y F2) cada uno de estos contiene 1000 páginas (P1...P1000 y P1001...P2000, respectivamente). Cada página contiene 100 registros identificados por p:i donde i es el slot al que pertenece el registro y p es la página. Un bloqueo de granualidad múltiple es utilizado con bloqueos S, X, IS, IX e SIX para cada una de las siguientes operaciones indique la secuencia de bloqueo que se debe realizar

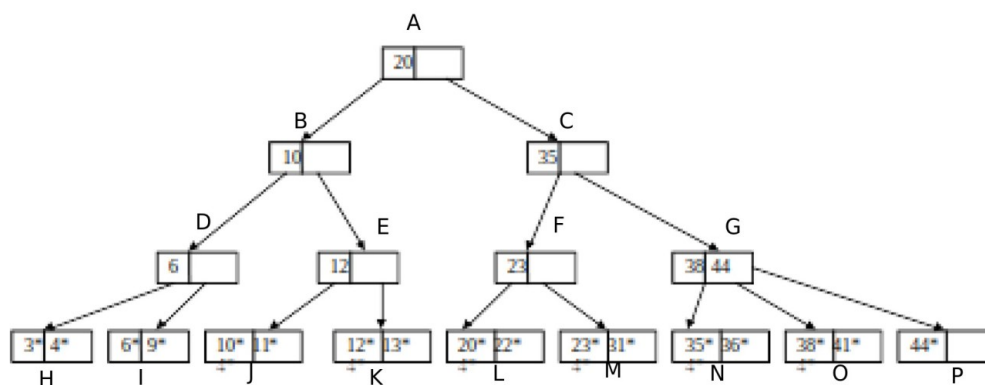
F1: 1 – 1000

F2: 1001 - 2000

- a) Leer de la página P2 a P1999  
S(D)
- b) Leer todas las paginas en F1 y modificar P1002  
IS(D)  
S(F1)  
IX(F2)  
X(P1002)
- c) Leer todas las paginas en F1 y modificar 10 páginas que únicamente pueden ser inidentificadas luego de leer F1  
IS(D)  
S(F1)  
IX(D)  
IX(F1)  
X(P1)  
X(P10)
- d) Borre todos los registros  
X(D)

#### Problema 5 (10 Puntos)

Considere el siguiente árbol



Describe los pasos necesarios para ejecutar cada una de las siguientes operaciones. Los incisos no son independientes.

- Buscar la entradas de datos 41\*

S(A)
S(C )
U(G)
S(G)
U(C )
S(N)
S(O)
U(G)
U(N)
U(O)

- Buscar todas las entradas de datos k\* con  $k \leq 20$

S(A)

S(B),

U(A)

S(D), S(E), U(B)

S(H), S(I), U(D)

S(J), S(K), U(E)

U(H), U(I), U(J), U(K)

- Insertar la entrada de datos 23\*

X(A)
X(C )
U(A)
X(F)
U©
X(M) → Se tendra que dividir ya que no tiene espacio

- Insertar la entradas de datos 7\*

X(A)
X(B )
U(A)
X(D)
U(B)

X(I) → Se tendra que dividir ya que no tiene espacio
--

- Insertar las entradas de datos 23\* y 75\*

X(A)
X(C )
U(A)
X(F)
U©
X(M) → Se tendra que dividir ya que no tiene espacio
U(F)
U(M)
X(C )
U(A)
X(G)
U(C )
X(P ) → Insertamos el 75