

1) OPTIMIZACIÓN

Criterio de Aprobación: Al menos 6,5 pts

Un canal de televisión mide la audiencia de sus programas y los de la competencia, y los guarda en un sistema con el siguiente esquema de base de datos:

Tablas

Audiencia

(fechahora, idPrograma, señal, idTipoPrograma, cantPersonas)

PK = {fechahora, idPrograma}

FK = {idTipoPrograma} referencia a Género.idTipoPrograma
 {idPrograma} referencia a Programa.idPrograma

Género

(idTipoPrograma, tipoPrograma)

PK = {idTipoPrograma}

FK = {}

Programa

(idPrograma, NombrePrograma)

PK = {idPrograma}

FK = {}

Consulta

```
SELECT A.fechahora, P.NombrePrograma, A.personas, A.Señal
FROM Audiencia A, Género G, Programa P
WHERE A.idTipoPrograma = G.idTipoPrograma
AND A.idPrograma = P.idPrograma
AND A.fechahora >= '20140612 17:00:00'
AND A.fechahora <= '20140630 19:00:00'
AND G.TipoPrograma = 'Deportivo'
```

Datos disponibles

- Índices:
 - I₁: B+ unclustered sobre señal en la tabla Audiencia
 - I₂: B+ unclustered sobre <fechahora, idPrograma> en la tabla Audiencia
 - I₃: B+ clustered sobre idPrograma en la tabla Programa
- Todos los índices B+ tienen altura 3
- Existen 40 bloques de memoria disponibles. El tamaño de bloque es 512 bytes.
- El primer campo de Audiencia ocupa 8 bytes; el resto, 4 bytes.
- El primer campo de Género ocupa 4 bytes; el segundo, 28 bytes.
- El primer campo de Programa ocupa 4 bytes; el segundo, 256 bytes.
- T_{GÉNERO} = 100
- Hay 200 señales que emiten su programación durante todo el día.
- Asumir que todos los punteros a hoja necesarios entran en una hoja del índice

El canal empezó a registrar y guardar la audiencia de cada día a partir del final del mundial de fútbol del 2010. En la tabla Audiencia se guarda la programación de todo el día de todas las señales desde que comenzó a medir audiencia. Por cada programa de cada día hay un registro en esta tabla. Se puede estimar que un programa dura 1 hora.

Las estadísticas de la tabla Programa están desactualizadas. Pero se sabe que durante un mundial de fútbol los programas deportivos conforman el 20% de la programación diaria.

- a. Armar el árbol canónico y aplicar **solamente** heurísticas algebraicas. No es necesario mostrar árboles intermedios (sí el inicial y final) pero se deben enumerar todas las heurísticas aplicadas.
- b. A partir del árbol anterior, armar un plan de ejecución (**puede** modificar el árbol de a)). **Justificar todas** las decisiones tomadas.
- c. Calcular el costo del plan anterior.

2) TRANSACCIONES

Criterio de aprobación: 1.a) 1, 1.b) 3, 1.c) 1.5, 1.d) 1.5, 2.a) 1.5, 2.b) 1.5. Se aprueba con 6.5

- 1) Dada la siguiente historia H, en el modelo READLOCK/WRITELOCK/UNLOCK sobre el conjunto de transacciones $\{T_1, T_2, T_3, T_4\}$ y el conjunto de items $\{X, Y\}$:

RL₃(X) RL₂(X) WL₃(Y) UL₃(X) WL₂(X) UL₃(Y) RL₄(Y) UL₂(X) RL₁(Y) RL₄(X) UL₁(Y) UL₄(X) WL₁(X)
UL₄(Y) UL₁(X) C₃ C₂ C₄ C₁

Asuma que entre un WL y su correspondiente UL ocurre solamente un Write del item (no ocurre un Read)

- a) ¿H es Legal? ¿Todas la T_i son 2PL? Justifique.
 b) Hacer el SG(H) e indicar si es serializable, y en caso afirmativo obtener todas las historias seriales equivalentes.
 c) ¿Cómo es H con respecto a recuperabilidad: ST, ACA, RC, (no RC)? Justifique.
 d) Idem anterior pero cambiando el orden de los commits por: C₄ C₁ C₂ C₃
- 2) Dada la siguiente historia en un planificador con *timestamp*

st₁, st₂, r₂(X), st₃, st₄, r₁(Y), r₄(Z), w₃(X), w₃(Y), w₄(Z), w₂(X), w₁(Y), r₃(Z)

en donde los valores iniciales de X, Y, Z son 0 y pasa que:

t₁ escribe Y = 1
 t₂ escribe X=2
 t₃ escribe X=3, Y=30, Z=300
 t₄ escribe Z = 4

- a) Decir que pasa en cada acción y que valores quedan en X,Y,Z si el planificador no usa *multiversión*
 b) Decir que pasa en cada acción X,Y,Z y que valores se tendrían para X,Y,Z si el planificador usa *multiversión*

3) LOGGING

Criterio de aprobación: a) 3, b) 3, c) 4. Se aprueba con 6.5

- a) Suponga que se desea iniciar un recovery luego de ocurrir un crash. Se dispone del siguiente undo/redo log con checkpointing no quiescente:

<START T₁>;<T₁,A,5,10>;<START T₂>;<START T₃>;<T₂,B,15,20>;<ABORT T₂>;<COMMIT T₁>;<T₃,C,25,30>;<START CKPT (T₃)>;<T₃,D,35,40>;<START T₄>;<T₄,E,45,50>;<END CKPT>;<COMMIT T₃>;<COMMIT T₄>

Indique justificando:

- i) Hasta dónde se debería examinar el log
 ii) Las transacciones a rehacer/deshacer
 iii) Los cambios a realizar en los items
 iv) Los cambios a realizar en el log
- b) Idem anterior pero el último registro del log es <COMMIT T₃>
 c) Idem anterior pero el último registro del log es <T₄,E,45,50>