

Recuperatorio 2do Parcial - 1er Cuatrimestre 2016 - Bases de Datos 6 de Julio 2016

- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con **nombre, apellido y LU**.
- Poner el **Nº de Orden** en la primer hoja de CADA EJERCICIO
- Cada tema tiene un criterio de aprobación propio y se recupera por tema. **NO DEBEN COMETERSE ERRORES CONCEPTUALES GRAVES.**
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.

1 NOSQL

Criterio de aprobación: se aprueba con 6.5 pts Con la siguiente escala: ítem a) 6.5 pts, el ítem b) 3.5 pts (debe tener al menos 1 punto en el ítem b)
Debe justificar los modelos elegidos.

- a) Una empresa provee acceso mediante tarjetas con lectores. Cada tarjeta tiene un identificador que es leído por los lectores para saber si tiene acceso en cada lugar. Cada puerta esta identificada con un número y cuenta con un lector. Si el identificador de tarjeta tiene permiso la puerta se abre y sino no. Para saber si un identificador tiene permiso los mismos se agrupan en roles. Los roles son los que tienen el permiso de acceso. Un identificador de tarjeta puede pertenecer a varios roles. Una vez que una persona entra en la recepción se le da una tarjeta y se activa la misma, cuando se retira de la empresa la devuelve en recepción y se desactiva.
 - i) Realizar un diseño para mantener el recorrido las tarjetas activas en una base clave-valor. Es decir debe guardarse la tarjeta con las puertas por las que pasó y por las que intento pasar. También debe saberse en que momento abrió o intento abrir una puerta.
 - ii) Realizar un diseño basado en documentos para modelar el problema. Puede realizar primero un modelo conceptual. Explicar fundamentadamente las decisiones de diseño tomadas. Cuando la persona que tiene la tarjeta se retira toda la información asociada a la tarjeta que está en la base clave valor (según ítem i) debe pasarse a la base de documentos y es eliminada de la base clave valor. Tome en cuenta que esa información es enormemente grande a lo largo del tiempo. No debería perder información al modelar los documentos.
- b) **Map-Reduce.** Suponga un documento que contiene identificación de tarjeta, fecha, una lista de puertas accedidas con su *timestamp* y una lista de puertas no accedidas (que intento pero no abrió) con su *timestamp*. Realice una consulta *map reduce* que devuelva la cantidad de accesos no permitidos y la cantidad de accesos permitidos por puerta durante el año pasado. O sea la puerta 1 tendrá por ejemplo 3000 accesos permitidos y 5000 no permitidos. El *reducer* debe ser combinable. Optimice el tráfico en red.

Justifique todas sus respuestas.

2 Concurrency and Recoverability

Criterio de aprobación: se aprueba con 6.5 pts. Valores parciales: a) 2 , b) 2 , c) 3 d) 3

Se tienen las siguientes transacciones:

$T_1 = wl_1(A); wl_1(D); u_1(A); u_1(D); c_1$

$T_2 = wl_2(C); wl_2(A); u_2(A); u_2(C); c_2$

$T_3 = rl_3(D); wl_3(C); u_3(D); u_3(C); c_3$

- Se pide realizar una Historia tal que se produzca un **deadlock**. Realice el grafo *wait-for* y diga cual transacción hay que abortar si se sigue una estrategia *Wait-Die*. Justificar.
- Para el mismo conjunto de transacciones realice una historia no recuperable. ¿Es la historia que construyo serializable? ¿Por qué?
- Asumiendo que en las transacciones dadas el conjunto de lectura está compuesto por los ítems que se bloquean para lectura y escritura, mientras que el conjunto de escritura sólo por los ítems que se bloquean para escritura realice una historia para un planificador basado en validación tal que se produzca un *rollback*. Justifique.
- Dado el siguiente archivo de Log cuyo *manager* utiliza una estrategia **Redo** con **checkpointing no-quietescente**:

- < START T_2 >
- < START T_1 >
- < $T_1, A, 100$ >
- < START T_3 >
- < $T_2, B, 90$ >
- < COMMIT T_1 >
- < $T_2, C, 44$ >
- < START CKPT (?, ?) >
- < START T_4 >
- < $T_4, D, 10$ >
- < $T_3, C, 50$ >
- < COMMIT T_3 >
- < $T_2, B, 200$ >
- < END CKPT >
- < COMMIT T_4 >
- < $T_2, E, 400$ >

Se pide

- Suponiendo una falla y que el último registro escrito es el 16 indicar (justificando) los pasos a seguir para la recuperación, indicando que debe hacerse con cada transacción, que modificaciones ocurren en el log si las hubiera, y los valores de los ítems modificados durante la recuperación, complete lo que esta con signos de pregunta.
- Idem el punto i) si el último registro escrito fuera el 13.

3 Optimización

Criterio de aprobación: se aprueba con 6.5 ptos, sin errores conceptuales, y el plan de ejecución propuesto en a), debe ser muy cercano al plan de menor costo posible.

Se cuenta con una base de datos de paquetes turísticos y fotografías de lugares turísticos, con las siguientes relaciones:

PAIS (idPais, nombre, continente)

TOUR (idTour, costo, descripción, empresa)

FOTOS (idFoto, idPais, idTour, descripción, año, valoración)

La tabla PAIS guarda países del mundo. La tabla TOUR guarda los tours disponibles. En FOTOS se guardan fotos de lugares turísticos. Las fotos tienen un país, un tour, un texto con una descripción detallada del lugar, el año de construcción si fuese una construcción, o el año en que fue sacada la fotografía, si no fuese una construcción (por ejemplo, un paisaje de la naturaleza), y una valoración, que es un número entero entre 1 y 10, que emerge del promedio que le otorgan los usuarios. Se desea optimizar la siguiente consulta:

```
SELECT T.idTour, F.idFoto
FROM FOTOS F, TOUR T, PAIS P
WHERE F.idTour = T.idTour AND F.idpais = P.idpais AND P.nombre = 'Francia' AND
F.valoración = 10 AND F.año < 600 AND F.año > (-900) AND T.costo < 10000
```

Se sabe adicionalmente que:

- Los atributos idX son numéricos enteros. El atributo costo es numérico flotante. Cantidad de bytes que ocupa cada atributo: idX (4), año (4), valoración (4), nombre(10), continente(10), costo (8), descripción (500), empresa (50)
- $T_{PAIS} = 100$; $T_{FOTOS} = 10.000.000$; $T_{TOUR} = 400$
- Los años de las fotos están entre -1000 y 2016.
- No hay dos países con el mismo nombre.
- Un tour cuesta como mínimo \$0 (gratis) y como máximo \$50000.
- Se dispone de los índices: **I1**: índice Hash sobre año en FOTOS; **I2**: índice Hash sobre costo en TOUR; **I3**: índice B+ clustered sobre (valoración, año) en FOTOS; **I4**: índice B+ unclustered sobre idPais en FOTOS
- Los índices B+ tienen una altura $X = 3$. En los índices Hash se asume un máximo de 10 bloques por bucket. Asuma que los punteros a tupla que se necesiten recorrer en los índices B+ entran en una hoja, y que una hoja entra en un bloque.
- Tamaño de un bloque $L_{bloque} = 2048$ bytes
- Cantidad de bloques en memoria: 10
- Si no hay especificación, suponga distribución uniforme.

Resuelva los siguientes items, justificando todas sus respuestas:

- Proponga un plan de ejecución optimizado para la consulta dada, indicando solamente el árbol final. Justifique cada una de las decisiones que tome para su propuesta de plan de ejecución. Justifique también porque decide utilizar o no utilizar cada uno de los índices disponibles.
- Calcule el costo de a) hasta el primer join inclusive.

Nota: En cada plan de ejecución utilizar materialización como input de una junta.