

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 14 de diciembre de 2022

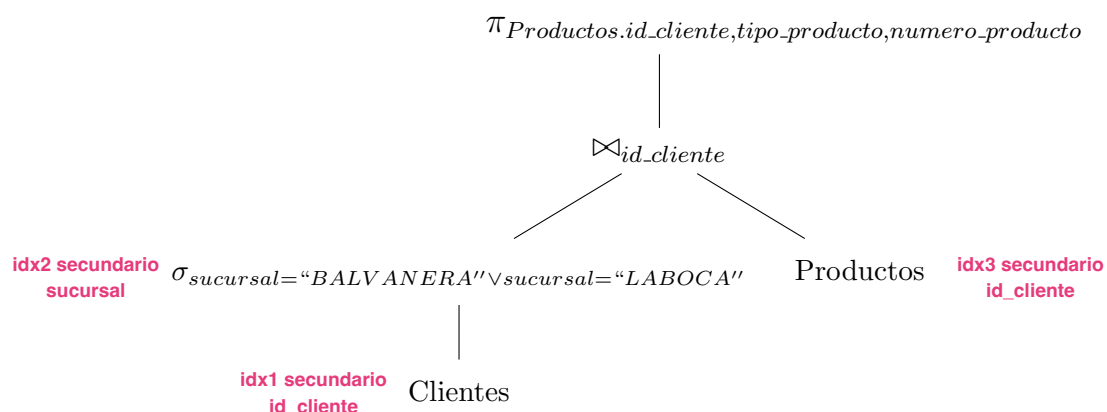
TEMA 20222C1						Padrón: _____ Apellido: _____ Nombre: _____ Cantidad de hojas: _____ <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente
Proc.		NoSQL		NoSQL		
SQL		CyT		Rec.		
Nota:						

Criterio de aprobación: El examen está compuesto por 6 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. Se aprueba con nota mayor o igual a 4(cuatro), equivalente a desarrollar el 60 % del examen correctamente.

1. (*Procesamiento de consultas*) La aseguradora *RISKE.ar* quiere analizar la información de los clientes de dos de sus sucursales más nuevas: “BALVANERA” y “LA BOCA”. Para ello ejecutará una consulta sobre las siguientes dos tablas,

- Clientes(id_cliente, nombre, apellido, email, sucursal)
- Productos(id_cliente, tipo_producto, numero_producto, fecha_contratacion)

que le devolverá los id's de los clientes de dichas sucursales y el tipo y número de productos que poseen:



A partir de dicho árbol de consulta, diseñe un plan de ejecución que tenga el menor costo estimado, y calcule el costo de dicho plan en términos de cantidad de accesos a disco.

Considere que dispone de infinita memoria, y de los siguientes 3 índices:

- Índice secundario idx1 sobre la columna id_cliente de *Clientes*, con altura 5.

- Índice secundario idx2 sobre la columna **sucursal** de *Clientes*, con altura 5.
- Índice secundario idx3 sobre la columna **id_cliente** de *Productos*, con altura 6.

y de la siguiente información de catálogo:

CLIENTES	PRODUCTOS
$n(\text{Clientes}) = 100.000$	$n(\text{Productos}) = 500.000$
$B(\text{Clientes}) = 10.000$	$B(\text{Productos}) = 25.000$
$V(\text{sucursal}, \text{Clientes}) = 500$	$V(\text{id_cliente}, \text{Productos}) = 100.000$
$H(I(\text{id_cliente}, \text{Clientes})) = 5$	$H(I(\text{id_cliente}, \text{Productos})) = 6$
$H(I(\text{sucursal}, \text{Clientes})) = 5$	

2. (*NoSQL*) En una base de datos en *MongoDB* son muy frecuentes dos tipos de consultas: una filtrando por un valor particular del atributo X cuya variabilidad es muy alta, y otra contando la cantidad de documentos con un valor particular del atributo Y, cuya variabilidad es muy baja. Suponiendo que la colección se desplegará sobre un sharding cluster, ¿qué estrategia utilizaría para que ambas consultas se respondan lo más rápido posible y cargando poco a la base de datos?

3. (*NoSQL*) Una base de datos distribuida almacena datos de tipo (*clave: valor*) replicados en 2 nodos. En un instante inicial t_0 , ambos nodos contienen la misma información:

(X: 2), (Y: 70), (Z: 8).

En ese momento, dos procesos P_1 y P_2 comienzan a ejecutar en forma concurrente los siguientes programas, que involucran operaciones de tipo *put()* y *get()* sobre el *data store*. El primer proceso opera sobre el nodo 1, mientras que el segundo lo hace sobre el nodo 2:

Proceso P_1 (Nodo 1)	Proceso P_2 (Nodo 2)
<code>put(Y, 6)</code>	<code>put(X, 9)</code>
<code>vX = get(X)</code>	<code>vY = get(Y)</code>
<code>vZ = get(Z)</code>	<code>put(Z, vY - 60)</code>

Si bien el *data store* se ocupa de propagar la información entre sus nodos, lo hace de manera asincrónica y sólo nos garantiza *consistencia eventual*. La siguiente figura ilustra los efectos que tuvo la ejecución concurrente de los dos procesos sobre el *data store*, indicando el valor de cada lectura y escritura.

P_1	W(Y)6	R(X)9	R(Z)10
P_2	W(X)9	R(Y)70	W(Z)10

$\xrightarrow{\text{Tiempo local}}$

Se pide:

- Indique si la ejecución presenta consistencia secuencial. Justifique su respuesta.
 - Indique si la ejecución es equivalente a alguna ejecución serial de los procesos (es decir si los procesos, vistos como transacciones, se ejecutaron de manera serializable). Justifique su respuesta.
4. (*SQL*) Para las mismas tablas del *Ejercicio 1*, escriba una consulta en SQL que encuentre, para cada sucursal, el nombre de el/los primer/os clientes (de acuerdo a la fecha) en contratar un producto de tipo ‘‘**SEGURO C/LAVA**’’. Las sucursales en que ningún cliente haya contratado aún ese tipo de producto no necesitan ser devueltas.

5. (*Concurrencia y transacciones*) Dado el siguiente solapamiento de transacciones:

$b_{T_1}; b_{T_2}; b_{T_3}; R_{T_1}(X); R_{T_2}(X); R_{T_2}(Y); W_{T_1}(Y); c_{T_1}; R_{T_3}(Z); W_{T_3}(X); R_{T_3}(Y); c_{T_2}; c_{T_3}$

Se pide:

- Dibuje el grafo de precedencias del solapamiento.
 - Indique si el solapamiento es serializable. Justifique su respuesta.
 - Indique si el solapamiento es recuperable. Justifique su respuesta.
6. (*Recuperación*) Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

```
01 (BEGIN, T1);
02 (WRITE T1, A, 10, 15);
03 (BEGIN, T2);
04 (WRITE T2, B, 20, 40);
05 (WRITE T1, C, 30, 35);
06 (BEGIN CKPT, T1, T2);
07 (WRITE T2, D, 40, 50);
08 (COMMIT, T2);
09 (WRITE T1, B, 40, 60);
10 (BEGIN, T3);
11 (WRITE T3, E, 15, 30);
12 (END CKPT);
13 (WRITE T3, D, 50, 15);
```

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.