

Base de Datos (75.15/95.05/TA044)

Segundo Parcial Promocional

TEMA 2024141	Proc. de Cons.	B	B	Fecha: 26 de junio de 2024
	NoSQL	B	B	Padrón: <u>109524</u>
	Conc. y Rec.	B	B	Apellido: <u>JUAN PULI</u>
Corrigió: <u>SEBAS</u> Nota: <u>10 (DIEZ)</u> <input checked="" type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente				Nombre: <u>MELINA</u> Cantidad de hojas: <u>6</u>

Criterio de aprobación: El examen está compuesto por 6 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. El examen se aprueba con nota mayor o igual a 4(cuatro) y la condición de aprobación es desarrollar un ítem bien (B/B-) en los 3 grupos de ejercicios (procesamiento de consultas, NoSQL, concurrencia/recuperación). Adicionalmente, no deberá haber más de dos ítems mal o no desarrollados.

1. (Procesamiento de consultas) La empresa multinacional *BueyStar* quiere obtener los clientes de dos países, utilizando la siguiente consulta:

▪ Clientes(id_cliente, nombre, email, pais, fecha_creacion, ...)

```
SELECT * FROM Clientes WHERE pais = 'Argentina' OR pais = 'Uzbekistan';
```

En la tabla existe un índice de clustering por la columna pais con una altura de 2, y además se cuenta con un histograma de los 3 valores más frecuentes y su cantidad de filas.

Se pide:

- a) Calcule, indicando el costo en acceso a bloques de disco para cada estrategia, si conviene utilizar el índice para resolver la consulta o si conviene efectuar un file scan de la tabla.
- b) Estime la cardinalidad del resultado de la selección en términos de cantidad de filas.

Considere para sus cálculos la siguiente información de catálogo:

Clientes	Histograma	
$n(\text{Clientes}) = 1.200.000$	Mexico	250.000
$B(\text{Clientes}) = 400.000$	Argentina	200.000
$V(\text{pais}, \text{Clientes}) = 83$	India	150.000

2. (Procesamiento de consultas) La plataforma de envíos *Lentti* guarda la información de los pedidos de sus usuarios en las siguientes tablas:

- Usuarios(id_usuario, email, direccion, ultimo_login)
- Pedidos(id_pedido, id_usuario, fecha, tipo)

Por un problema de auditoria, precisa saber en qué fechas un usuario hizo pedidos urgentes con la siguiente consulta SQL:

```
SELECT fecha FROM Usuarios u, Pedidos p
WHERE p.id_usuario = u.id_usuario
AND u.email = 'immcnabb@nfl.com' AND p.tipo = 'URGENTE';
```

La tabla de usuarios cuenta con un índice por id_usuario (I1) y otro por email (I2). La tabla de pedidos cuenta con un índice por id_usuario (I3) y otro por tipo (I4). Todos son índices secundarios.

Se pide:

- a) Genere un árbol de consulta para una resolución eficiente de la consulta.
- b) Calcule el costo de resolver la consulta con el plan que surge de dicho árbol de consulta.

Para resolver ambos items se cuenta con la siguiente metadata:

Usuarios	Pedidos
$n(\text{Usuarios}) = 80.000$	$n(\text{Pedidos}) = 320.000$
$B(\text{Usuarios}) = 2.000$	$B(\text{Pedidos}) = 32.000$
$V(\text{email}, \text{Usuarios}) = 80.000$	$V(\text{id_usuario}, \text{Pedidos}) = 80.000$
	$V(\text{tipo}, \text{Pedidos}) = 10$
$\text{Height}(I1) = 4$	$\text{Height}(I3) = 4$
$\text{Height}(I2) = 4$	$\text{Height}(I4) = 1$

3. (NoSQL - MongoDB) El sitio de publicaciones científicas *Paper View* guarda en una base de datos Mongo los datos de los papers publicados con la siguiente estructura de documento:

```

1 {
2   "id": 10910355903998401931,
3   "titulo": "Base de Datos, de la B a la D",
4   "autores": ["Mariano Villani", "Alejandro John"],
5   "categoria": "Informatica",
6   "puntaje": 4.2
7 }
```

Lo que buscan es obtener información sobre los autores de papers que pertenezcan a la categoría "Informática": para cada autor que haya publicado al menos 10 de esos papers, quieren conocer la cantidad de esos papers publicados y el promedio de puntaje entre ellos, con la siguiente estructura :

```

1 {
2   "autor": "Mariano Villani",
3   "cantidad": 30,
4   "promedio_puntaje": 5.3
5 }
```

- Escriba una consulta en MongoDB que devuelva el listado según las condiciones indicadas.
 - Explique por qué atributos puede shardearse la colección de papers para que la resolución de la consulta sea lo más distribuida posible. En caso de que haya atributos por los que shardear haga la resolución menos distribuida, indique cuales son con una breve explicación del por qué.
4. (Neo4j) La famosa red social *LinkedOut* está sufriendo un ataque de trolls! Ha detectado que muchos usuarios se estan organizando para poner puntajes altos a ciertas publicaciones. La información la tiene almacenada en una base de datos en Neo4j con los siguientes nodos y aristas:

```

1 (us1:Usuario {username: 'conejo'})
2 (us2:Usuario {username: 'aguantemessi'})
3 (pub1:Publicacion {titulo: 'Developer SSSSr', id: '7097321', contenido:
4   'Se busca estudiante avanzado....'})
5 (pub2:Publicacion {titulo: 'Scrum Master', id: '4032123', contenido: '
6   Reconocida empresa busca....'})
7 ...
8 (us1)-[:PUNTUA {puntaje: 10}]->(pub1)
9 (us2)-[:PUNTUA {puntaje: 9}]->(pub1)
10 (us1)-[:PUNTUA {puntaje: 10}]->(pub2)
11 (us2)-[:PUNTUA {puntaje: 10}]->(pub2)
```

Para detectar un par de trolls, busca que ambos hayan puntuado con un puntaje de 8 o mas a al menos 5 publicaciones. Además es necesario que no haya una publicación en la que uno dio un puntaje de 8 o más y el otro la haya puntuado con un 7 o menos. Escriba una consulta en Cypher (lenguaje de consulta de Neo4j) que devuelva, sin repetir, los pares de usuario que cumplan con esas condiciones anteriores.

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

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

- 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 REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

01 (BEGIN, T1);	09 (BEGIN, T4);
02 (WRITE, T1, A, 10);	10 (COMMIT, T3);
03 (BEGIN, T2);	11 (WRITE, T4, C, 17);
04 (WRITE, T2, D, 20);	12 (END CKPT);
05 (BEGIN, T3);	13 (COMMIT, T1);
06 (COMMIT, T2);	14 (BEGIN CKPT, T4);
07 (BEGIN CKPT, T1, T3);	15 (WRITE, T4, B, 30);
08 (WRITE, T3, B, 40);	16 (COMMIT T4);

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando:

- Hasta qué punto del archivo de log se deberá retroceder.
- Qué cambios deberán ser realizados en disco y en el archivo de log.

ME LIMA

JUAN G. FLORES



B

① Datos:

- Índice de clustering columna PAIS $\frac{14}{14} = 2$ ✓
 - $F(c) = \frac{1.200.000}{400.000} = 3$ ✓

B

② Con Índice:

Como la query hace selección por OR entre los valores el costo de obtener el apellido cliente de Argentina y luego de aquellos clientes de Venezuela.

COSTO-ARG: Como se sabe la cantidad de filas exactas de clientes de Argentina:
 acceso a los datos por índice

$$\text{COSTO-ARG} = \frac{14}{2} + \left\{ \frac{\text{CANT-FILAS-ARG}}{F(c)} \right\} = 2 + \left\{ \frac{200.000}{3} \right\} = 66,669$$

66667

COSTO-VEN: Como no está en el interfaz luego extraer la cantidad de filas por cliente

$$\text{FILAS-VEN} = \frac{\text{FILAS-clientes} - (\text{FILAS-MEXICO}) - (\text{FILAS-ARG}) - (\text{FILAS-INDIA})}{[V(\text{PAIS, clientes}) - 3]}$$

3 PAISES
 en histograma

$$= \frac{1.200.000 - [230.000 + 200.000 + 130.000]}{(33-3)}$$

$$= 7500 \quad \checkmark$$

$$\text{Costo-UEs} = H + \left[\frac{\text{Fijas-UEs}}{F(C)} \right] = 2 + \left[\frac{2500}{3} \right] = 250.2$$

$$\text{Costo-TOTAL} = \text{Costo-Arg} + \text{Costo-UEs} = 66669 + 2502 = 69171 \quad \checkmark$$

~~A~~

5) $\text{CANT-Fijas-TOTALES} = \text{CANT-UEs} + \text{CANT-Arg} = 7500 + 200.00 = 2107500 \quad \checkmark$

B

~~A~~ $\text{Costo-File-SCN} = B(\text{Clientes}) = 400.000 \quad \checkmark$

Se puede observar por separado uno de los ítems, la guerra y el miembro más eficiente. ✓

MELINA

JAVIER

(2)

(2) Datos:

- Como la $V(email, usuarios) = 80.000 = m(usuarios)$ estoy considerando que la relación por email no es buena - mucho más que por pedidos_tipo

(1)

select i.fidena

pipeline.

(3)

where p.tipo = "urgente"

pipeline.

(2)

join p.id_usuario = u.id_usuario

pedidos

(1)

where u.email = x

USUARIOS

u

(1)

Seleccion por cada segundo.

$$\checkmark \text{Costo-1} = H(12) + \left\lceil \frac{m(usuarios)}{V(email, usuarios)} \right\rceil = 4 + 1 = 5$$

(2)

Junta por metodo de unico loop \Rightarrow índice constante.
Como suponemos que el tiempo obtenido del paso 1. mas los para por pipeline:


$$\begin{aligned} \text{Costo-2} &= B(\text{PASO-1}) + A(\text{PASO-1}) \cdot \left(H(13) + \left\lceil \frac{n(p)}{V(id-usuario, p)} \right\rceil \right) \\ &= \left\lceil \frac{m(usuarios)}{V(email, usuarios)} \right\rceil \cdot \left(4 + \left\lceil \frac{320.000}{80.000} \right\rceil \right) = \\ &= 1 \cdot (4 + 4) = 8 \end{aligned}$$

- (3) Como se hace por primera vez luego el costo es 0 ya que no se evalúa la condición en memoria.

$$\text{COSTO} - 3 = 0$$

- (4) Como la prefijación NO tiene Distinto luego, como también se hace por primera vez el costo es 0.

$$\text{COSTO} - 4 = 0$$

$$\begin{aligned}\text{COSTO} - \text{TOTAL} &= \text{COSTO} - 1 + \text{COSTO} - 2 + \text{COSTO} - 3 + \text{COSTO} - 4 = \\ &= 5 + 8 + 0 + 0 = 13\end{aligned}$$


③ MEDIA SENSITIVE

③ PIPELINE = [

```

    ② {
      "MATCH" : {
        "category" : "INFORMATICA" ✓
      }
    },
    {
      "FILTER" : "$AUTHORS" ✓
    },
    {
      "$group" : {
        "_id" : "$AUTHORS",
        "COUNT" : { "$sum" : 1 },
        "median-price" : { "$avg" : "$PRICE" } ✓
      }
    },
    {
      "MATCH" : {
        "COUNT" : { "$gt" : 10 } ✓
      }
    },
    {
      "PROJECT" : {
        "_id" : 0,
        "AUTHOR" : "$_id",
        "COUNT" : 1,
        "median-price" : 1 ✓
      }
    }
  ]

```

Collection, Aggregate (pipeline)

(4) M5 L5A same for

B

(4) MATCH $(U1: usuario) \rightarrow [P1: Puntuat] \rightarrow (P: Puntuat) \leftarrow [P2: Puntuat] - (U2: usuario)$

Where $P1.Puntuat \geq 8$ AND $P2.Puntuat \geq 8$ AND

NOT EXISTS

$(U1) \rightarrow [P3: Puntuat] \rightarrow [P: Puntuat] \leftarrow$

$\leftarrow [P4: Puntuat] - (U2)$

Where $(P3.Puntuat \geq 8$ AND $P4.Puntuat \leq 7)$ OR

$(P4.Puntuat \geq 8$ AND $P3.Puntuat \leq 7)$

} AND $U1.username \leq U2.username$.

With $U1, U2, COUNT(DISTINCT P) \geq CWT-P$

Where $CWT-P \geq 5$

Return $U1.username, U2.username$.

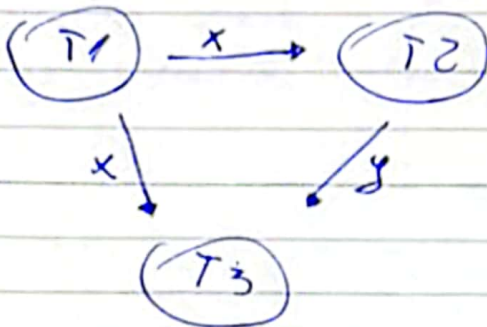
B

(S)

CONFLICTOS:

- $WT_1(x), RT_3(x) \Rightarrow T_1 \xrightarrow{x} T_3$ ✓
- $WT_1(x), RT_2(x) \Rightarrow T_1 \xrightarrow{x} T_2$ ✓
- $RT_2(y), WT_3(y) \Rightarrow T_2 \xrightarrow{y} T_3$ ✓

(2)



(b) Como el prefijo de precedencia \overline{NO} tiene ciclos, entonces es irresoluble

Forma: $T_1 \rightarrow T_2 \rightarrow T_3$

(c) DEFINICION de recuperabilidad:

Un momento es recuperable si y solo si ninguna transacción T resalta al commit hasta tanto todas las transacciones que escriben datos antes de que T los haya leído sean committeadas.

Juego el momento si es recuperable por que T_1 no lee ningún dato modificado por otra transacción, T_2 lee y pone antes de su committeado a T_1 T_3 por el dato \rightarrow

5) PARA disminuir LA CONSULTA LO MÉS POSIBLE SE SIGUE LOS SIGUIENTES PASOS:

✓ 1. NO elegir valores por lo que no filtre contra uno solo

✓ 2. NO elegir rondas por valores que pueda ser erróneo ya que no es posible porque no siempre

NO SERÁ
Bueno
Interpretar
POR
AUTORES

3. NO elegir rondas por valores de poca variedad lo que hace que caigan en pocos modos

✓ 4. Dependiendo del Intencional, no rondas por algo que pueda tener un desbalance en los datos.

Seguendo estas recomendaciones, luego para esta consulta en especial lo más eficiente será rondas por el ID ya que la distribución será uniforme con una buena función de hash.

Aun así, si la cantidad de autores, los títulos de papers es grande, luego también será una buena opción.

✓ Por otro lado, los puntajes solo serán buenos si la variedad es grande, de lo contrario no será un buen atributo por el cual rondar.

✓ Lo que sí es NO eficiente rondar en este query y por categorías (ya que todos los de información van a un mismo modo) haciendo así la distribución más uniforme.

modificado por $T_1(x)$ pero este ultimo
contiene entre $pul = 3$ ^(al igual que $pul = 2$), asegurando el
solapamiento recuperable.

② y lee x pero este contiene entre pul ✓

(6)

MELWA Here Jan

- 13 (6) Se encuentra primero en BEGIN CKPT. (04)
Como el checkpoint solo llegó hasta
este punto no nos sirve, y entonces deberemos
ir a buscar un checkpoint anterior al
log (07) ✓

Como en el BEGIN CKPT están actuando
las transacciones $\{T_1, T_3\}$ luego, aunque
hayan completado no se nos asegura
que en cualquier momento puedan guardarse
a disco. Por esto mismo, debe reírse
todas aquellas transacciones que
completaron después del BEGIN CKPT (07) ✓
(no incluye T_2)

Transacciones a reírse: $\{T_1, T_3, T_4\}$

(*)

- $A = 10$ // por T_1 línea 02
- $B = 40$ // por T_3 línea 08
- $C = 17$ // por T_4 línea 11
- $D = 30$ // por T_4 línea 15 ✓

- (*) luego debe retroceder hasta la línea 01
por lo donde comienza T_1 . ✓

Modificaciones en disco:

luego de reírse las operaciones,
se debe hacer backup a disco de
estos datos recuperados.

Como no hay transacciones que no
hayan completado, luego no se
escriben ningún abort.

REDO:

Antes de realizar el commit, todo nuevo valor y enmendado por la transacción debe ser sobrescrito en el log, en disco

No obliga a guardar el item modificado en disco antes de commitar la transacción SÓLO en el momento de log. No existe en este algoritmo el item o actualizado en disco luego de commitar la transacción.