Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Primer Semestre de 2019

IIC 2413 – Bases de Datos
Interrogación 3

Pregunta 1: Índices, algoritmos y transacciones

a) [2 ptos] En clases explicamos el funcionamiento del *B+Tree Index*, pero nunca mencionamos cómo funciona este índice en caso de valores duplicados. Por ejemplo, considera la tabla `Persona(id int, nombre VARCHAR(100), edad int)`. Es natural pensar que la edad pueda tener valores repetidos para distintas personas. Explica cómo podríamos hacer que un *B+Tree unclustered* pueda indexar el atributo `edad` de la relación `Persona`. Mencione cómo cambian los costos de busqueda para consultas de igualdad y rango (no es necesario que de valores concretos, pero si que entregue la intuición). **Hint**: piense en las overflow pages del *Hash Index*.

b) [2 ptos] Considera la relación `Persona(id int PRIMARY KEY, nombre varchar(100), edad int)`. Por alguna razón, en esta relación la primary key está indexada con un B+Tree *unclustered*, mientras que la edad está indexada con un Hash Index *clustered*. La relación tiene 1 millón de tuplas y la columna `id` posee los valores del 0 al 999999. En cada página caben 50 tuplas y 200 punteros. El árbol es de altura h y las hojas están ocupadas al 75 %. El Hash Index tiene una página por *bucket* (i.e. es suficientemente grande y no tiene *overflow pages*), y cada edad va a parar a un *bucket* distinto.

Indique el costo en I/O de las siguientes consultas:

1. `SELECT * FROM Persona WHERE id=4444`
2. `SELECT * FROM Persona WHERE nombre='Claudio Bravo'`
3. `SELECT * FROM Persona WHERE edad=33`
4. `SELECT * FROM Persona WHERE id<10000`
5. `SELECT * FROM Persona WHERE edad>=25 and edad<35`

c) [2 ptos] Considera las siguientes relaciones:

- `R(a int, b int)`
- `S(b int, c int)`

Quieres hacer la consulta $R \bowtie S$, pero las relaciones no están indexadas. Sin embargo, en clases vimos un algoritmo de ordenamiento. Explica:

- **[1 pto]** De qué nos serviría ordenar previamente ambas relaciones por el atributo `b`.
- **[1 pto]** El costo total de computar la consulta, considerando el costo de haber ordenado `R` y `S`.

Bonus [2 décimas en la interrogación] En clases un alumno expuso una teoría de por qué Arturo Prat había sido elevado a héroe nacional a pesar de haber perdido el Combate Naval de Iquique. Explique esta teoría brevemente.

Pregunta 2: Transacciones y Logs

Transacciones

a) [1 pt] Considere el Schedule del cuadro 1. Indique para cada valor de X (puede ser R o W) si el *schedule* es *conflict serializable* o no. Explique por qué.

T1	T2	T3	T4
R(a)			
	W(a)		
	R(b)		
		W(b)	
		W(c)	
			W(c)
			R(d)
X(d)			

Cuadro 1: schedule problema 2.

b) [2 pts] De una explicación de alto nivel de por qué al usar *Strict-2PL* sólo podemos generar *schedules* de tipo *conflict serializable*. **Hint**: Piense en las aristas que se pueden generar gracias a los posibles tipos de *locks*.

Redo Logging

Suponga que su sistema tuvo una falla. Al reiniciar el sistema, el sistema se encuentra con el *log file* que se muestra a continuación, en la tabla "Log Redo". Suponiendo que la política de *recovery* es la de *Redo Logging*, indique:

a) Desde qué parte del *log* debo comenzar el proceso de *redo*.

b) Qué variables deben rehacer sus cambios y cuál es el valor con el que quedarán.

c) Qué variables (de las que aparecen en el *log*) no son cambiadas en el proceso.

d) Si no hubiesemos encontrado la línea `<END CKPT>`, ¿desde qué parte del *log* debería comenzar el proceso de *redo*?. Justifique su respuesta explicando qué significa no haber encontrado un `<END CKPT>`.

Log Redo
<START T1>
<T1, a, 1>
<COMMIT T1>
<START T2>
<T2, b, 10>
<T2, c, 20>
<START T3>
<T3, a, 15>
<START CKPT (T2, T3)>
<T3, d, 23>
<START T4>
<END CKPT>
<COMMIT T3>
<T4, e, 11>

Pregunta 3: MongoDB

Uno de tus compañeros de universidad que dio Bases de Datos el semestre anterior te recomendó para trabajarn en la red social *Fotograma*[1]. Esta red social utiliza una base de datos MongoDB que posee un modelo bastante simple:

- Existe una colección de usuarios con datos básicos que toda red social debe tener.
- Existe una colección de fotogramas. Estas son las imágenes que los usuarios suben a la red social. Entre otros atributos, se destaca la presencia de la URL de la imagen y la descripción, que corresponde a un texto.
- Cada documento que corresponde a la colección de usuarios tiene una lista con los `id` de los fotogramas a los que ha marcado con "me gusta".
- Existe una colección de comentarios, en donde cada documento contiene el id del usuario que emitió el comentario, el id del fotograma comentado y el texto del comentario

Un ejemplo de documentos que se pueden encontrar en la base de datos MongoDB son los siguientes:

```
// Usuarios
{ "uid": 23,
  "nombre": "Arturo Vidal",
  "bio": {
    "trabaja_en": "FC Barcelona", "edad": 33, "descr": "Jugador de Chile y FC Barcelona"
  },
  "me_gusta": [3, 5, 6] }

// Fotogramas
{ "fid": 1,
  "posteada_por": 23,
  "fotograma": "fotograma.com/foto/12345.jpg",
  "descripción": "El tiempo pasa y las verdaderas amistades perduran!! #MalaOnda" }

// Comentarios
{ "fid": 1,
  "uid": 24,
  "comentario": "Oye Arturo, no hay pa' qué" }
```

En tu primer día de empresa te encargaron diseñar algunas consultas. Dado que en el curso de bases de datos aprendiste MongoDB a la perfección, esto no va a ser un problema para ti. Utilizando PyMongo o JavaScript se pide un procedimiento para cada una de las consultas a continuación. Si necesitas un índice por texto debes indicar el comando para crearlo.

- **[2 pts]** Entregue el *id* de fotograma, junto al link del fotograma, junto al número de comentarios que tiene para cada fotograma que contiene el texto "#Cobreloa" y que no tenga el texto "Cobreloa perdió" en su descripción de fotograma.
- **[2 pts]** Entregue el *id* del fotograma, junto al número total de "Me Gusta" de todas las fotos que hayan sido comentadas por el usuario con *id* 23.
- **[2 pts]** Entregue todos los documentos de la colección Fotograma (se debe entregar el documento con todos sus atributos) que no tengan comentarios con la frase "Vamos Chile Bicampeón".

[1] Sí, es una mezcla de Fotolog con Instagram.