IIC2413 –Bases de Datos

Ayudantía I3

1. Resumen

- El principal costo en la base de datos es la lectura a disco, es lo que quiero minimizar.
- Tiempo de total de lectura en disco es `Tiempo de busqueda + Delay Rotacional + Tasa de transferencia`.
- En índices, **Search Key** es el parámetro a buscar, **Data Entry** es la información almacenada.
- **Clustered Index** es un índice que está ordenado de la misma forma que los records en la DB, **Clustered File** es un índice en que el *data entry* contiene el record, **Unclustered Index** índice que no es clustered.
- El costo en I/O del Hash Index es: $\frac{|DataEntries|}{B \cdot N}$. B es el número de records por página, N el número de páginas.
- El costo de I/O del B+ Tree es: $O(log_{\frac{B}{2}}(\frac{2 \cdot Tuplas}{B}))$.
- El costo de I/O del External Merge Sort optimizado es: $4 \cdot N$ asumiendo un buffer óptimo de tamaño $B \geq \sqrt{N}$. N es el número de páginas de la tabla a ordenar.
- El costo de I/O del Block Nested Loop Join es: $Costo(R) + (Pags(R)/Buffer)Costo(S)$.
- Un schedule es serializable si el grafo de precedencia es acíclico.

2. Ejercicios Ayudantía

2.1. Preguntas breves

- ¿En qué momento me conviene usar un hash index? ¿En qué momento me conviene usar un B+ Tree?
- Indique la diferencia entre un índice clustered y unclustered
- Defina las propiedades **ACID**.
- Defina un caso en el que le convenga usar MongoDB y otro en el que no. Defina un caso en el que le convenga usar una base de datos columnar y otro en el que no.

2.2. Problemas

Problema 1) Sea la relación $R(a, b, c, d)$ cuyo tamaño es de 2 millones de tuplas, en que cada página contiene en promedio P tuplas. Las tuplas de R están ordenados de manera aleatoria (cuando no hay índice). El atributo a es además un candidato a llave primaria, cuyos valores van del 0 al 1.999.999 (distribuídos uniformemente). Para cada una de las consultas a continuación, diga el número de I/O que se harán en cada uno de los siguientes casos:

- Analizar R sin ningún índice.
- Usar un *B+Tree Unclustered* sobre el atributo a. El árbol es de altura h. La cantidad de punteros por pagina es de M. Las hojas están ocupadas al 70 %.
- Usar un *Hash Index Clustered* con B buckets. Suponga que la función de hash actúa de manera uniforme.

Las consultas son:

T1	T2	T3	T4
W(a)			
	R(b)		
			R(b)
		W(b)	
R(b)			
	W(a)		

Cuadro 1: My caption

1. Encontrar todas las tuplas de R.
2. Encontrar todas las tuplas de R tal que $a < 70$.
3. Encontrar todas las tuplas de R tal que $a = 207890$.
4. Encontrar todas las tuplas de R tal que $a > 50$ y $a \leq 200$.

Comente sus resultados e indique que índice prefiere en los distintos casos.

Problema 2) Suponga que tiene una relación `Personas(id, nombre, edad)`. La llave primaria está indexada con un B+Tree clustered de altura h y con las hojas ocupadas al 100 %. La relacción tiene 500.000 tuplas y caben 100 por página. Se pide que indique:

- Número de fases y costo en I/O de para la consulta `SELECT * FROM Personas ORDER BY edad`.
- Costo en I/O de la consulta `SELECT * FROM Personas ORDER BY id`.

Si utiliza External Merge Sort, puede usar la versión optimizada. Asuma buffer suficientemente grande. Comente sus resultados.

Problema 3) Sea el schedule del cuadro 1, determine si es *Conflict Serializable*. ¿Qué pasa si se estuviera utilizando *Strict 2PL*?

Problema 4) Considere un esquema de dos tablas `A(id int, name varchar(10))` y `B(id int, name varchar(10))`. Suponga que quiere hacer la consulta en álgebra relacional $\pi_{A.name,B.name}(A \bowtie B)$ pero solamente dispone de un archivo que se ve de la siguiente forma:

```
A,1,palabra1
A,2,palabra2
A,3,palabra3
B,1,palabra1
...
```

El primer término antes de la coma representa la tabla, el segundo el id y el tercero el name. Entregue un algoritmo Map - Reduce que ejecute la consulta deseada.

Problema 5) Piense en una base de datos en MongoDB con dos colecciones, una de autores y otra de cuentos:

```
// Personas
{
  "uid": 1,
  "name": "Gabriel García Márquez",
}

// Cuentos
{
  "rid": 1,
  "uid": 1,
  "cuento": "El rastro de su sangre en la nieve",
  "type": "Realismo Mágico",
  "content": "... El médico no le dio la importancia que Billy Sánchez esperaba.
                'Hizo bien en decírmelo,' dijo, y se fue detrás de la camilla... "
}
```

Entregue una serie de pasos relacionados a MongoDB para obtener todos los autores de cuentos de tipo Realismo Mágico que en su contenido mencionan la palabra "Soledad" pero no la frase "Cien Años". Dado que la base de datos es muy grande, debe usar búsqueda por texto. En concreto debe crear un índice, entregar la consulta de búsqueda por texto y luego hacer el join según lo visto en clases.