Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2413 - Bases de Datos

Taller 8 - Índices

Fecha: 28 de mayo de 2025

1^{er} semestre 2025 - Profesores: Eduardo Bustos - Christian Álvarez

---

Pregunta 1

Dada la tabla de datos Clientes conformada por 1.000 tuplas

Cliente	**Email**	**Teléfono**	**Dirección**	**Comuna**
Agustin Matte	Agustin.Matte@bbdduc.utopia	56911904638	Calle Estado 456	Santiago
Agustin Reyes	Agustin.Reyes@bbdduc.utopia	56972317149	Calle Prat 789	Iquique
...	...	...	...	...

donde ... indica que hay más datos en la tabla Clientes.

I Indique un posible esquema para la tabla Clientes (cliente, email y teléfono son atributos únicos y no nulos).

```
Solución: Clientes(cliente varchar(30) unique not null,
mail varchar(100) unique not null, telefono int unique not null,
direccion varchar(100), comuna varchar(30))
```

II Defina un surrogate key llamada Key para la tabla Clientes como llave primaria e inclúyala en el esquema.

```
Solución: Clientes(key serial PRIMARY KEY,
cliente varchar(30) unique not null,
mail varchar(100) unique not null, telefono int unique not null,
direccion varchar(100), comuna varchar(30))
```

III Defina un índice para la surrogate key llamada Key-index y proponga una posible función de Hash para un factor de carga 1

Solución: CREATE INDEX key-index ON Clientes(key), También es válido que digan que la PK ya está indexada

F(key)= key mod 100

IV Defina un índice secundario para el atributo email llamado mail-index. ¿Si la función de hash es f(mail)=Largo del string (mail) que efecto tendría sobre el factor de carga?

Solución: CREATE INDEX mail-index ON Clientes(mail)
La función Largo(mail) es mínimo 4 y máximo 100 y puede estar muy concentrada, por lo que podría haber mucho overflow.

V Construya tres diagramas, uno para la tabla y uno para cada uno de los dos índices con 10 registros por bucket

Tabla:

1	Ra
2	Rb
..	....
n	Rc

Key-Index:

1	Ra, Ri,....,Rj
2	Rk,Rt,.....,Rz
..	....
100	Ru, Rv,....,Rw

Mail-Index:

1	Ra, Ri,....,Rj
2	Rk,Rt,.....,Rz
..	....
n	Ru, Rv,....,Rw

VI ¿Cómo haría Ud. para que la tabla de Hash mail-index no repita cada tupla completa de la tabla Clientes?

Solución: El registro puede contener el valor de la llave y un puntero al bucket del otro índice o un puntero al registro de la tabla

Pregunta 2

Sea la relación $R(a, b, c, d)$ cuyo tamañoo es de 1 millón de tuplas, en que cada página contiene P tuplas. Las tuplas de R están ordenadas de manera aleatoria. El atributo a es además un candidato a llave primaria, cuyos valores van del 0 al 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 y cada página contiene M punteros ($M > P$).
- Usar un $B+Tree$ clustered sobre el atributo a. El árbol es de altura h y cada página de hoja está ocupada al 60 %.
- Usar un $Hash\ Index$ unclustered con 1 millón de buckets. Cada página del índice contiene M punteros ($M > P$).
- Usar un $Hash\ Index$ clustered con 1 millón de buckets.

Las consultas son:

I Encontrar todas las tuplas de R.

II Encontrar todas las tuplas de R tal que $a < 50$.

III Encontrar todas las tuplas de R tal que $a = 50$.

IV Encontrar todas las tuplas de R tal que $a > 50$ y $a < 100$.

Sin índice

Para el caso cuando no se tiene ningún índice, para todas las consultas debemos acceder a todas las páginas de disco donde se almacenan las tuplas. Si tenemos 10^6 tuplas, y capacidad P en cada página, necesitamos $\frac{10^6}{P}$ páginas para almacenar toda la información. Por lo tanto, para todas las consultas tendremos un costo I/O de $\frac{10^6}{P}$.
,

B+Tree Clustered

I En este caso, para encontrar todas las tuplas, primero tenemos que bajar por el árbol hasta la primera hoja (esto suma un costo I/O de h). Ahora, nos dicen que cada página está ocupada en un 60 % por lo que la cantidad de páginas que utilizaremos para guardar todas las tuplas es $\frac{10^6}{0{,}6*P}$. Cómo ya tenemos la primera página (al bajar hasta la primera hoja), le restamos 1 a la suma, dando como resultado: $(h-1)+\frac{10^6}{0{,}6*P}$

II Como queremos las tuplas donde $a < 50$, la cantidad de tuplas que tenemos son 50 (recordemos que los valores empiezan en 0). Con esto, la cantidad de páginas donde están almacenadas estas 50 tuplas, siguiendo la lógica de la consulta anterior son $\frac{50}{0{,}6P}$. Aquí también debemos sumar el costo de bajar por el árbol de h, y el -1 por tener la primera página necesaria al bajar por el árbol. El resultado es: $(h-1)+\frac{50}{0{,}6*P}$

III Como queremos la tupla donde $a = 50$, solo debemos bajar por el árbol (h). El resultado es: h

IV La idea es similar a la consulta 2, sólo que en vez de 50 tuplas, tendremos 49 (que son la cantidad de tuplas entre 50 y 100 no incluídos). Entonces, se suma el costo de bajar por el árbol (h) y la cantidad de páginas $\frac{49}{0,6*P}$. Se resta -1 por la misma razón explicada anteriormente. El resultado es: $(h-1)+\frac{49}{0,6*P}$

B+Tree Unclustered

I Para los casos unclustered, sabemos que cada página tiene M punteros a páginas de disco. Entonces, la cantidad de páginas para almacenar todas las tuplas será $\frac{10^6}{M}$ (ya que $10^6 = paginas * M$). En el caso de obtener todas las tuplas, el costo I/O será el bajar por el árbol (h), más la cantidad de páginas a recorrer ($\frac{10^6}{M}$), más la cantidad de tuplas (10^6). En el factor $\frac{10^6}{M}$ ya está incluída la primera página qué obtuvimos bajando por el árbol (-1). El resultado es: $(h-1)+\frac{10^6}{M}+10^6$

II Se sigue la misma lógica que para la consulta anterior, pero considerando solo 50 tuplas (entre 0 y 49). El resultado es: $(h-1)+\frac{50}{M}+50$

III En el caso de obtener un valor particular, debemos bajar por el árbol (h) hasta la página con el puntero, y luego acceder a la tupla (+1). El resultado es: $h+1$

IV La misma lógica que para las dos primera consultas, pero considerando solo las 49 tuplas (entre 51 y 99). El resultado es: $(h-1)+\frac{49}{M}+49$

Hash Index Clustered

I Como tenemos 10^6 buckets, no tendremos colisiones (cada uno corresponde a una tupla). Si queremos acceder a todas las tupla de R, vamos a tener que "hacer una llamada O(1)" por cada tupla, lo que sería costo I/O de 10^6

II La misma lógica aplica en esta consulta, pero con 50 tuplas. El resultado es: 50

III Aquí simplemente tenemos que acceder al valor, por lo que el costo sería 1.

IV La misma lógica que consultas anteriores, pero con 49 tuplas. El costo es: 49

Hash Index Unclustered

I Cuando queremos todas las tuplas, vamos a tener que ir a buscar la página del índice (+1) y luego la página indicada por el puntero (+1), esto para cada tupla. Entonces el costo será $2 * 10^6$

II Seguimos la misma lógica de la primera consulta, pero con 50 tuplas ($2 * 50 = 100$).

III La misma lógica pero con una sola tupla ($2 * 1 = 2$).

IV La misma lógica de la primera consulta, pero con 49 tuplas ($2 * 49 = 98$).