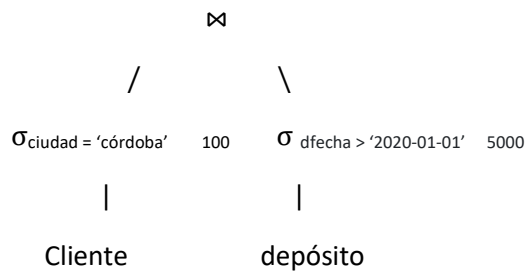


### Ejercicio 1:

Árbol binario de ejecución:



Factor de selectividad:

- $Fs(\text{ciudad} = \text{'Córdoba'}, \text{cliente}) = 1/(V(\text{ciudad}, \text{cliente})) = 1/100$
- $Fs(\text{dfecha} > \text{'2020-01-01'}, \text{depósito}) = (\max(\text{dfecha}, \text{depósito}) - \text{'2020-01-01'}) / (\max(\text{dfecha}, \text{deposito}) - \min(\text{dfecha}, \text{depósito})) = (\text{'2020-12-31'} - \text{'2020-01-01'}) / (\text{'2020-12-31'} - \text{2010-01-01'}) = 365/10 * 365 = 1/10$ .
- $Fs(\text{cliente.cid} = \text{depósito.cid}, \text{cliente}, \text{depósito}) = 1/\max(V(\text{cid}, \text{cliente}), V(\text{cid}, \text{depósito})) = 1/\max(100000, 100000) = 1/100000$

Deposito equi-distribuidos por cliente y hay 100000 clientes, entonces cada cliente tiene en promedio 10 registros por depósito.

Decidir operadores físicos: \_

- Uso algoritmo para selección con índice secundario e igualdad en no clave candidata.
- Uso algoritmo para selección con índice primario para mayor.
- Para reunión natural usar el algoritmo: algoritmo de merge-sort híbrido

Tamaño en bloques de las tablas de la BD: clientes 10000 y depósito 50000

Calculo de tamaño de resultados intermedios en bloques.

$$|\sigma_{\text{ciudad} = \text{'córdoba'}} \text{ cliente}| = |\text{cliente}| * fs(\text{ciudad} = \text{'Córdoba'}, \text{cliente}) = 100000 * 1/100 = 1000$$

Entran 10 registros de cliente por bloque, luego necesito 100 bloques para este resultado intermedio.

$$|\sigma_{\text{dfecha} > \text{'2020-01-01'}} \text{ depósito}| = |\text{depósito}| * fs(\text{dfecha} > \text{'2020-01-01'}, \text{depósito}) = 1000000 * 1/10 = 100000$$

Entran 20 registros de depósito por bloque, luego necesito 5000 bloques para este resultado intermedio.

Ahora calculamos el costo de los operadores físicos.

#### **Algoritmo de selección con índice secundario e igualdad en no clave candidata:**

Costo =  $(h_i + n)$  transferencias de bloques, donde  $h_i$  altura de árbol B+ y  $n$  numero de registros recogidos.

Altura del árbol B+ =  $\lceil \log_{\lceil n/2 \rceil}(K) \rceil$  donde  $K$  cantidad de valores de la clave de búsqueda en la tabla. Y  $n$  cantidad de entradas por nodo.

$K = 100$ . Supongamos que una ciudad ocupa 50 B, un puntero 6B. Un bloque tiene 4 KiB. Esto da 73 entradas por bloque. Así  $n = 73$ .

$$\text{Altura del árbol B+} = \lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_{\lceil 73/2 \rceil}(100) \rceil = \lceil \log_{37}(100) \rceil = \lceil \log(100)/\log(37) \rceil = \lceil 2/1,56 \rceil = \lceil 1,28 \rceil = 2.$$

Hay aproximadamente 1000 registros para Córdoba.

La estimación es 1002 transferencias de bloques.

### **Algoritmo para selección con índice primario para mayor**

Usar índice para encontrar primer registro con fecha mayor a la dada y escanear la tabla secuencialmente desde allí.

Costo:  $hi + b$  transferencias, donde  $b$  numero de bloques con registros con  $dfecha > '2020-01-01'$

Hay 100000 registros de esos, y 5000 bloques.

Cantidad de valores de  $K = 1000000$ .

Asumimos que entran 20 entradas por bloque.  $N=20$ . (es razonable – 20 registros por bloque y monto ocupa como un puntero.)

$$\text{Altura del árbol B+} = \lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_{\lceil 20/2 \rceil}(1000000) \rceil = \lceil \log_{10}(1000000) \rceil = \lceil 6 \rceil = 6.$$

Luego la estimación es: 5006 transferencias de bloques.

### **Algoritmo de join merge sort:**

$\sigma_{ciudad = 'córdoba'}$  cliente ordenada por cid.

$\sigma_{dfecha > '2020-01-01'}$  depósito hay que ordenarla por cid.

Vamos a usar ordenamiento externo. En memoria entran 12 bloques.  $br = 5000$

$$\begin{aligned} \text{Num transf bloques} &= br(2 \lceil \log_{M-1}(br/M) \rceil + 1) = 5000(2 \lceil \log_{12-1}(5000/12) \rceil + 1) \\ &= 5000(2 \lceil \log_{11} 416 \rceil + 1) = 5000(2 \lceil \log_{10} 416 / \log_{10}(11) \rceil + 1) = 5000(2 \lceil 2,61 / 1,04 \rceil + 1) \\ &= 5000(2 \lceil 2,5 \rceil + 1) = 5000 * 7 = 35000. \end{aligned}$$

A esto hay que sumar  $br + bs = 100 + 5000$ .

En total da: 40100 transferencias de bloques.

## Ejercicio 2:

Para los conjuntos de una tabla, los tamaños, costos y mejores planes son:

	{Persona}	{Bibliotecario}	{trabajaEn}
Size	1000	400	1500
Cost	0	0	0
Best plan	persona	bibliotecario	trabajaEn

Ahora consideramos pares de tablas. El costo para cada uno es 0 porque no hay tablas intermedias en la reunión de dos tablas (las dos tablas son chicas y están en memoria principal). Hay dos posibles planes. Tomamos la menor de las tablas como el argumento izquierdo de la expresión de reunión natural.

$$|\text{Bibliotecario} \bowtie \text{persona}| = |\text{Bibliotecario}| * |\text{persona}| * 1/\max(V(\text{bibliotecario}, \text{DNI}), V(\text{persona}, \text{DNI})) = 400 * 1000 / \max(400, 1000) = 400$$

$$|\text{Bibliotecario} \bowtie \text{trabajaEn}| = |\text{Bibliotecario}| * |\text{trabajaEn}| * 1/\max(V(\text{bibliotecario}, \text{DNI}), V(\text{trabajaEn}, \text{DNI})) = 400 * 1500 * 1/\max(400, 750) = 400 * 1500 / 750 = 800$$

$$|\text{Persona} \bowtie \text{trabajaEn}| = |\text{persona}| * |\text{trabajaEn}| * 1/\max(V(\text{persona}, \text{DNI}), V(\text{trabajaEn}, \text{DNI})) = 1000 * 1500 * 1/\max(1000, 750) = 1000 * 1500 / 1000 = 1500$$

	{persona, bibliotecario}	{bibliotecario trabajaEn}	{trabajaEn, persona}
Size	400	800	1500
Cost	0	0	0
Best plan	Bibliotecario $\bowtie$ persona	Bibliotecario $\bowtie$ trabajaEn	Persona $\bowtie$ trabajaEn

Ahora consideramos la tabla de reuniones para 3 tablas. La única manera es tomar dos tablas para reunir primero. La estimación de costo es el tamaño de la tabla intermedia. Como queremos que este costo sea tan pequeño como sea posible consideramos cada par de las 3 tablas y tomamos el par con menor tamaño usando la tabla previa: esto nos da el par {persona, bibliotecario}. Considerando esta par como el argumento del natural join obtenemos:

	{persona, bibliotecario}, {trabajaEn}	{bibliotecario, trabajaEn}, {persona}	{trabajaEn, persona}, {bibliotecario}
Cost	400 + 1500 = 1900	800 + 1000 = 1800	1500 + 400 = 1900

$$|(\text{bibliotecario} \bowtie \text{trabajaEn}) \bowtie \text{persona}| = |(\text{bibliotecario} \bowtie \text{trabajaEn})| * |\text{persona}| * 1/\max(V(\text{bibliotecario} \bowtie \text{trabajaEn}, \text{DNI}), V(\text{persona}, \text{DNI})) = 800 * 1000 * 1/1000 = 800$$

agrupamiento	{Bibliotecario, persona, trabajaEn}
Size	800
Cost	1800
Best plan	(bibliotecario $\bowtie$ trabajaEn) $\bowtie$ persona

Entonces la consulta se puede reescribir a:

$\Pi_{\text{nombre, DNI, nombreBib}} (\sigma_{\text{antigüedad} > 5 \text{ and rol} = \text{'bibliotecario'}} ((\text{bibliotecario} \bowtie \text{trabajaEn}) \bowtie \text{persona}))$

Meto la primera condición del select en bibliotecario

$\Pi_{\text{nombre, DNI, nombreBib}} (\sigma_{\text{rol} = \text{'bibliotecario'}} (\sigma_{\text{antigüedad} > 5} \text{ bibliotecario} \bowtie \text{trabajaEn}) \bowtie \text{persona}))$

Meto la primera condición del primer select en trabajaEn

$\Pi_{\text{nombre, DNI, nombreBib}} ((\sigma_{\text{antigüedad} > 5} \text{ bibliotecario} \bowtie \sigma_{\text{rol} = \text{'bibliotecario'}} \text{ trabajaEn}) \bowtie \text{persona}))$

Empujo proyecciones para eliminar atributos.

$((\Pi_{\text{DNI}} (\sigma_{\text{antigüedad} > 5} \text{ bibliotecario})) \bowtie (\Pi_{\text{DNI, nombreBib}} \sigma_{\text{rol} = \text{'bibliotecario'}} \text{ trabajaEn})) \bowtie \Pi_{\text{nombre, DNI}} \text{ persona})$

La reunión es más profunda a la izquierda, por lo tanto cumple con heurística.