Procesamiento y Optimización de Consultas

Ejercicios Simples

Bases de Datos

2017

Pasos para resolver un ejercicio de optimización

1. Entender la consulta
2. Pasar de SQL a Algebra Relacional
3. Armar árbol canónico
4. Aplicar heurísticas algebraicas
5. Armar plan de ejecución estimando costos
6. Calcular costos

Resumen de Costos

Tipo de archivo / índice	Costo de exploración completa	Costo de búsqueda por igualdad (A = k)	Costo de búsqueda por rango ($k_1 <= A <= k_2$)
Heap file	B_R	B_R	B_R
Sorted file	B_R	$\log_2(B_R) + \lceil T' / FB_R \rceil$	$\log_2(B_R) + \lceil T' / FB_R \rceil$
Índice B+ clustered sobre A	-	$X_I + \lceil T' / FB_R \rceil$	$X_I + \lceil T' / FB_R \rceil$
Índice B+ unclustered sobre A	-	$X - 1 + \lceil T' / FB_I \rceil + T'$	$X - 1 + \lceil T' / FB_I \rceil + T'$
Índice hash estático sobre A	-	$MBxB_I + T'$	-

- **T'** es la cantidad de tuplas que cumplen con el criterio de la búsqueda
- FB_R**I** es el factor de bloqueo del archivo
- FB_I es el factor de bloqueo del índice I
- $MBxB_I$ es la cantidad máxima de bloques de un bucket

Datos

Idea de la clase

Calcular el costo de consultas pequeñas y simples, con un mismo esquema, cambiando de entorno para ver cómo influyen en los accesos a disco.

Esquema a utilizar

Medico(mId, nombre, especialidad, fechaIngreso)

Datos

- tamaño de bloque: 4096 bytes
- longitud de cada campo: 128 bytes
- cantidad de tuplas: 1000

Algunos cálculos:

- **longitud tupla M:** $L_M = 128 bytes \times 4(registros) = 512 bytes$
- **factor de bloqueo de M:**
  $FB_M = \lceil tam\_bloque / L_M \rceil = 4096 bytes / 512 bytes = 8 tuplas / bloque$
- **cantidad de tuplas:** $T_M = 1000 tuplas$
- **cantidad de bloques:**
  $B_M = \lceil T_M / FB_M \rceil = \lceil 1000 tuplas / 8 tuplas / bloque \rceil = 125 bloques$

Escenario 1: selección por rango usando HeapFile

```
SELECT *   FROM Médico M
    WHERE "1/1/90" < fechaIngreso
    AND fechaIngreso < "1/1/91"
```

Escenario 1: selección por rango usando HeapFile

```
SELECT *   FROM Médico M
    WHERE "1/1/90" < fechaIngreso
    AND fechaIngreso < "1/1/91"
```

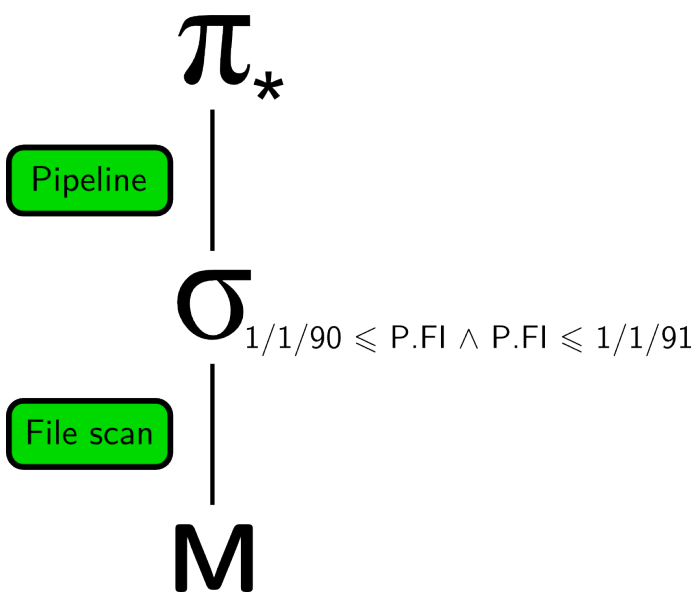
En álgebra relacional:

$$\pi_{*}(\sigma_{\text{"}1/1/90\text{"} < fechaIngreso < \text{"}1/1/91\text{"}}(M))$$

Escenario 1: selección por rango usando HeapFile (cont.)

$$\pi_*(\sigma_{\text{"}1/1/90\text{"} < \mathit{fechaIngreso} < \text{"}1/1/91\text{"}}(M))$$

Árbol canónico con su plan de ejecución:



Escenario 1: selección por rango usando HeapFile (cont.)

- La relación M está organizada internamente como un HeapFile.

Escenario 1: selección por rango usando HeapFile (cont.)

- La relación M está organizada internamente como un HeapFile. Por lo tanto sus registros están desordenados.

Escenario 1: selección por rango usando HeapFile (cont.)

- La relación M está organizada internamente como un HeapFile. Por lo tanto sus registros están desordenados.
- Entonces para ver el costo de una selección por rango hay que recorrer el archivo completo linealmente (file scan).

Escenario 1: selección por rango usando HeapFile (cont.)

- La relación M está organizada internamente como un HeapFile. Por lo tanto sus registros están desordenados.
- Entonces para ver el costo de una selección por rango hay que recorrer el archivo completo linealmente (file scan).
- **Costo de búsqueda:** $B_M = 125$ bloques.

Escenario 2: selección por rango usando árbol B^+ clustered

Datos:

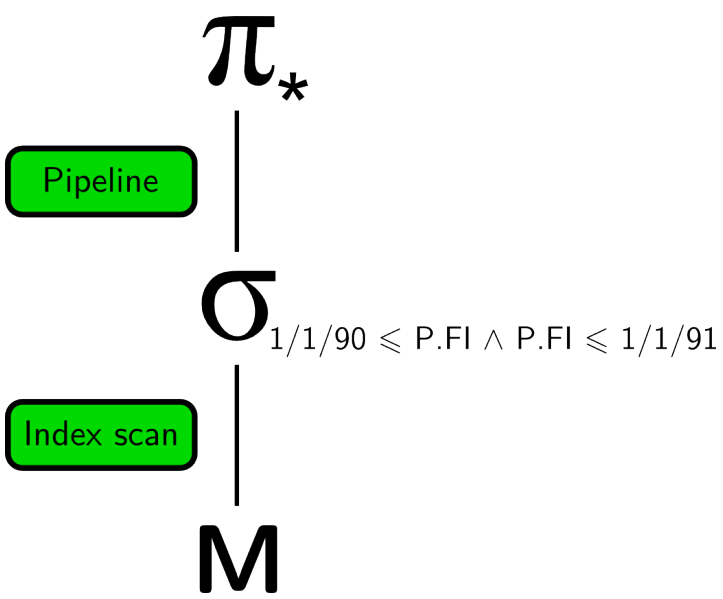
- Misma consulta.
- Costo de acceso al índice de 3 bloques.
- Indice sobre el atributo fechaIngreso.
- Asumimos que entre el '1/1/90' y el '1/1/91' ingresaron 200 médicos.

Escenario 2: selección por rango usando árbol B^+ clustered

Datos:

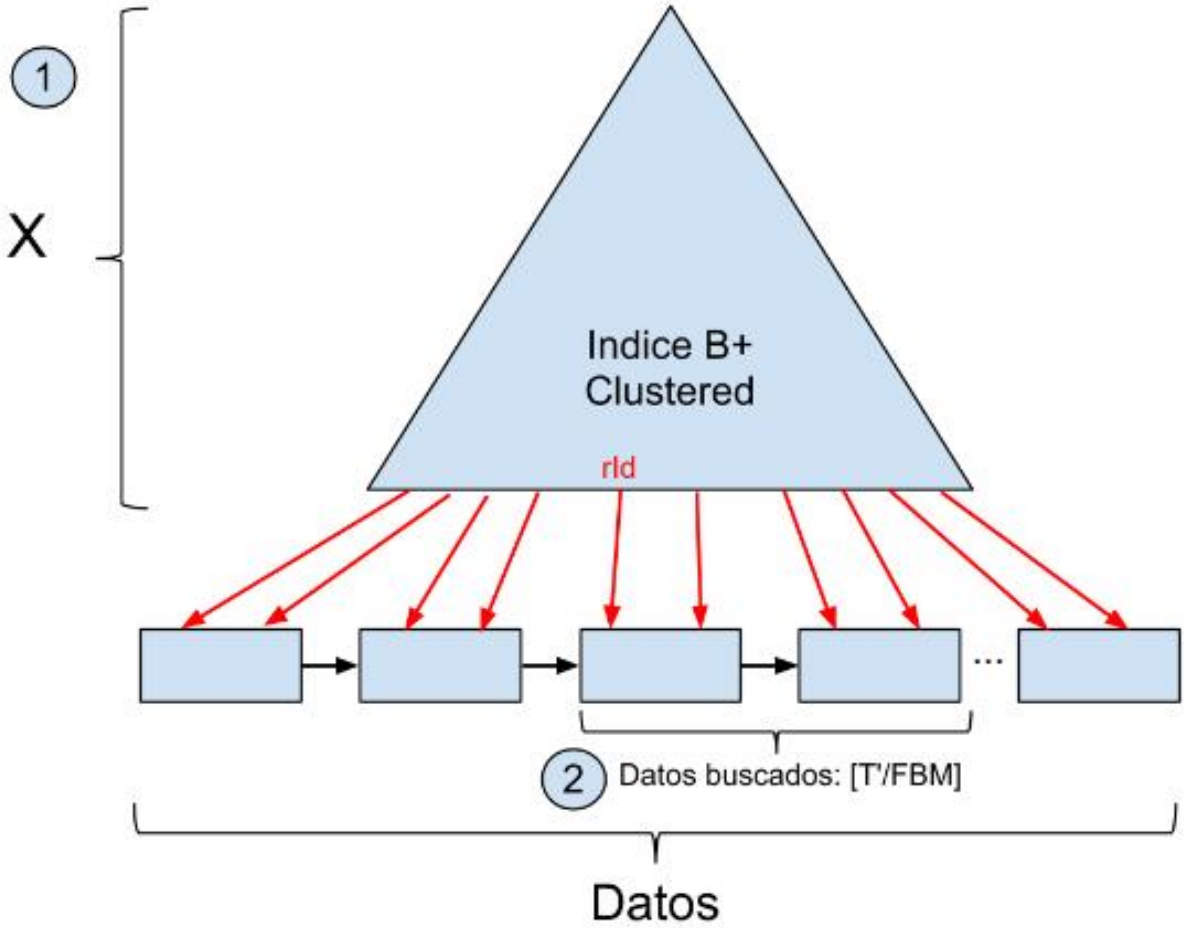
- Misma consulta.
- Costo de acceso al índice de 3 bloques.
- Indice sobre el atributo fechaIngreso.
- Asumimos que entre el '1/1/90' y el '1/1/91' ingresaron 200 médicos.

Como tenemos un índice que coincide con la clave de búsqueda cambia el plan de ejecución:



Escenario 2: selección por rango usando árbol B^+ clustered

Costos? Recordemos que...



Escenario 2: selección por rango usando árbol B^+ clustered (cont.)

Pasos a seguir para calcular el costo:

Escenario 2: selección por rango usando árbol B^+ clustered (cont.)

Pasos a seguir para calcular el costo:

- usamos el índice clustered sobre fechaIngreso,

Escenario 2: selección por rango usando árbol B^+ clustered (cont.)

Pasos a seguir para calcular el costo:

- usamos el índice clustered sobre fechaIngreso,
- recorremos el índice en busca del primer valor que cumpla con el criterio de seleccíon, bajando niveles en el árbol hasta llegar a la hoja donde se encuentra el puntero al primer bloque de datos (costo: **3 bloques**),

Escenario 2: selección por rango usando árbol B^+ clustered (cont.)

Pasos a seguir para calcular el costo:

- usamos el índice clustered sobre fechaIngreso,
- recorremos el índice en busca del primer valor que cumpla con el criterio de seleccíon, bajando niveles en el árbol hasta llegar a la hoja donde se encuentra el puntero al primer bloque de datos (costo: **3 bloques**),
- leemos en orden (los datos están ordenados físicamente de acuerdo a fechaIngreso) (costo: $\lceil T'/FB_M \rceil$ = [200 tuplas/8 tuplas/bloque] = **25 bloques**,

Escenario 2: selección por rango usando árbol B^+ clustered (cont.)

Pasos a seguir para calcular el costo:

- usamos el índice clustered sobre fechaIngreso,
- recorremos el índice en busca del primer valor que cumpla con el criterio de seleccíon, bajando niveles en el árbol hasta llegar a la hoja donde se encuentra el puntero al primer bloque de datos (costo: **3 bloques**),
- leemos en orden (los datos están ordenados físicamente de acuerdo a fechaIngreso) (costo: $\lceil T'/FB_M \rceil$ = [200 tuplas/8 tuplas/bloque] = **25 bloques**,
- costo total: $\lceil T'/FB_M \rceil + X$ o sea **28 bloques**

Escenario 3: selección por igualdad usando HashTable

```
SELECT *
    FROM Médico M
    WHERE especialidad = "traumatología"
```

Datos del ejercicio:

- índice HashTable sobre el atributo especialidad,
- el $MBxB_i$ = cantidad máxima de bloques que ocupa un bucket del índice i (basado en hashing) es 4
- asumimos que hay 100 traumatólogos

Escenario 3: selección por igualdad usando HashTable

```
SELECT *
    FROM Médico M
    WHERE especialidad = "traumatología"
```

Datos del ejercicio:

- índice HashTable sobre el atributo especialidad,
- el $MBxB_i$ = cantidad máxima de bloques que ocupa un bucket del índice i (basado en hashing) es 4
- asumimos que hay 100 traumatólogos

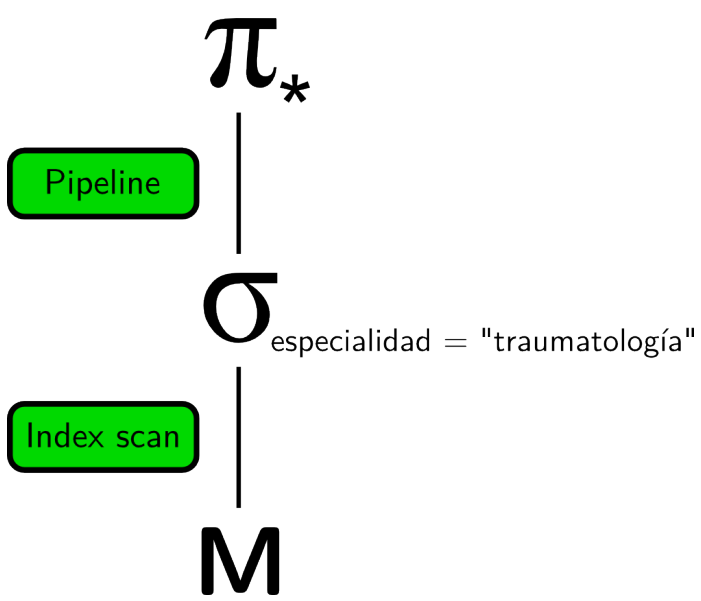
Pasamos la consulta a álgebra relacional:

$$\pi_*(\sigma_{especialidad=\text{"}traumatologia\text{"}}(M))$$

Escenario 3: selección por igualdad usando HashTable (cont.)

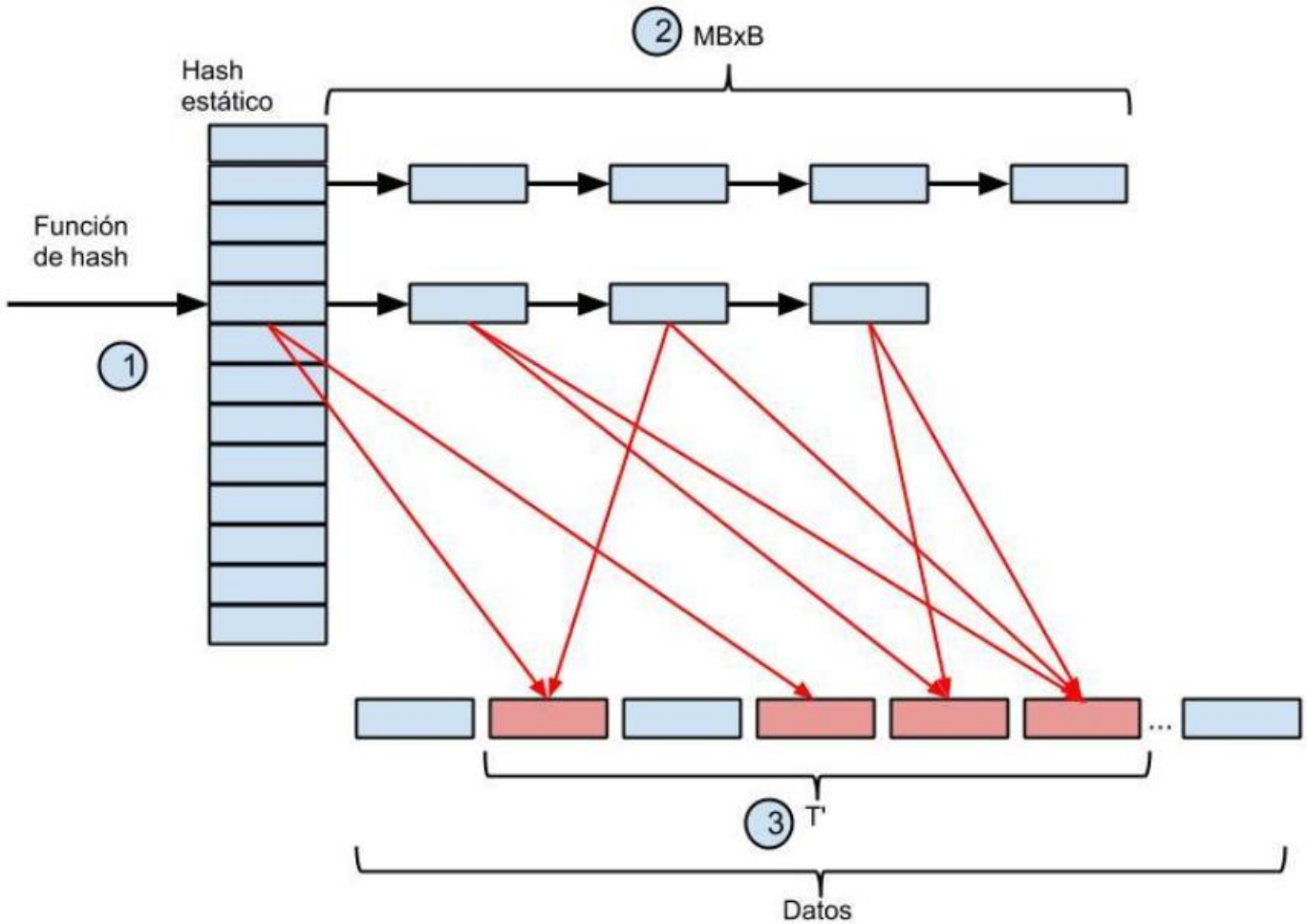
$$\pi_*(\sigma_{especialidad=\text{"}traumatologia\text{"}}(M))$$

Árbol canónico:



Escenario 3: selección por igualdad usando HashTable (cont.)

La tabla de hash era:



Escenario 3: selección por igualdad usando HashTable (cont.)

Pasos a seguir para calcular el costo:

Escenario 3: selección por igualdad usando HashTable (cont.)

Pasos a seguir para calcular el costo:

- identificar el bucket correspondiente (asumimos **costo 0**),

Escenario 3: selección por igualdad usando HashTable (cont.)

Pasos a seguir para calcular el costo:

- identificar el bucket correspondiente (asumimos **costo 0**),
- recorrer todos los bloques del bucket para encontrar los punteros que se corresponden con las tuplas que coinciden con la clave buscada: $MBxB_i$ en peor caso (**costo 4**),

Escenario 3: selección por igualdad usando HashTable (cont.)

Pasos a seguir para calcular el costo:

- identificar el bucket correspondiente (asumimos **costo 0**),
- recorrer todos los bloques del bucket para encontrar los punteros que se corresponden con las tuplas que coinciden con la clave buscada: $MBxB_i$ en peor caso (**costo 4**),
- acceder tantos bloques de datos como tuplas busquemos, para lo cual hay que leer todos los bloques apuntados por el bucket. Peor caso: cada bloque apunta a un bloque nuevo (**costo 100**),

Escenario 3: selección por igualdad usando HashTable (cont.)

Pasos a seguir para calcular el costo:

- identificar el bucket correspondiente (asumimos **costo 0**),
- recorrer todos los bloques del bucket para encontrar los punteros que se corresponden con las tuplas que coinciden con la clave buscada: $MBxB_i$ en peor caso (**costo 4**),
- acceder tantos bloques de datos como tuplas busquemos, para lo cual hay que leer todos los bloques apuntados por el bucket. Peor caso: cada bloque apunta a un bloque nuevo (**costo 100**),
- **costo total: 104**

Escenario 4: selección por rango usando HashTable (cont.)

Si tenemos de nuevo la consulta del primer ejercicio y el índice hash?

```
SELECT *
    FROM Médico M
    WHERE "1/1/90" < fechaIngreso AND fechaIngreso < "1/1/91"
```

Escenario 4: selección por rango usando HashTable (cont.)

Si tenemos de nuevo la consulta del primer ejercicio y el índice hash?

```
SELECT *
    FROM Médico M
    WHERE "1/1/90" < fechaIngreso AND fechaIngreso < "1/1/91"
```

El índice no sirve, hay que recorrer el archivo completo. **El índice hash solo sirve para búsquedas por igualdad.**

Índice Hash y Rango

Para usar el ínidice por hash habría que tener todos los valores de búsqueda y en general podrían ser infinitos.

Escenario 5: cuando todo está en el índice

Tenemos la consulta:

```
SELECT fechaIngreso
    FROM Médico M
    WHERE especialidad = "traumatología"
```

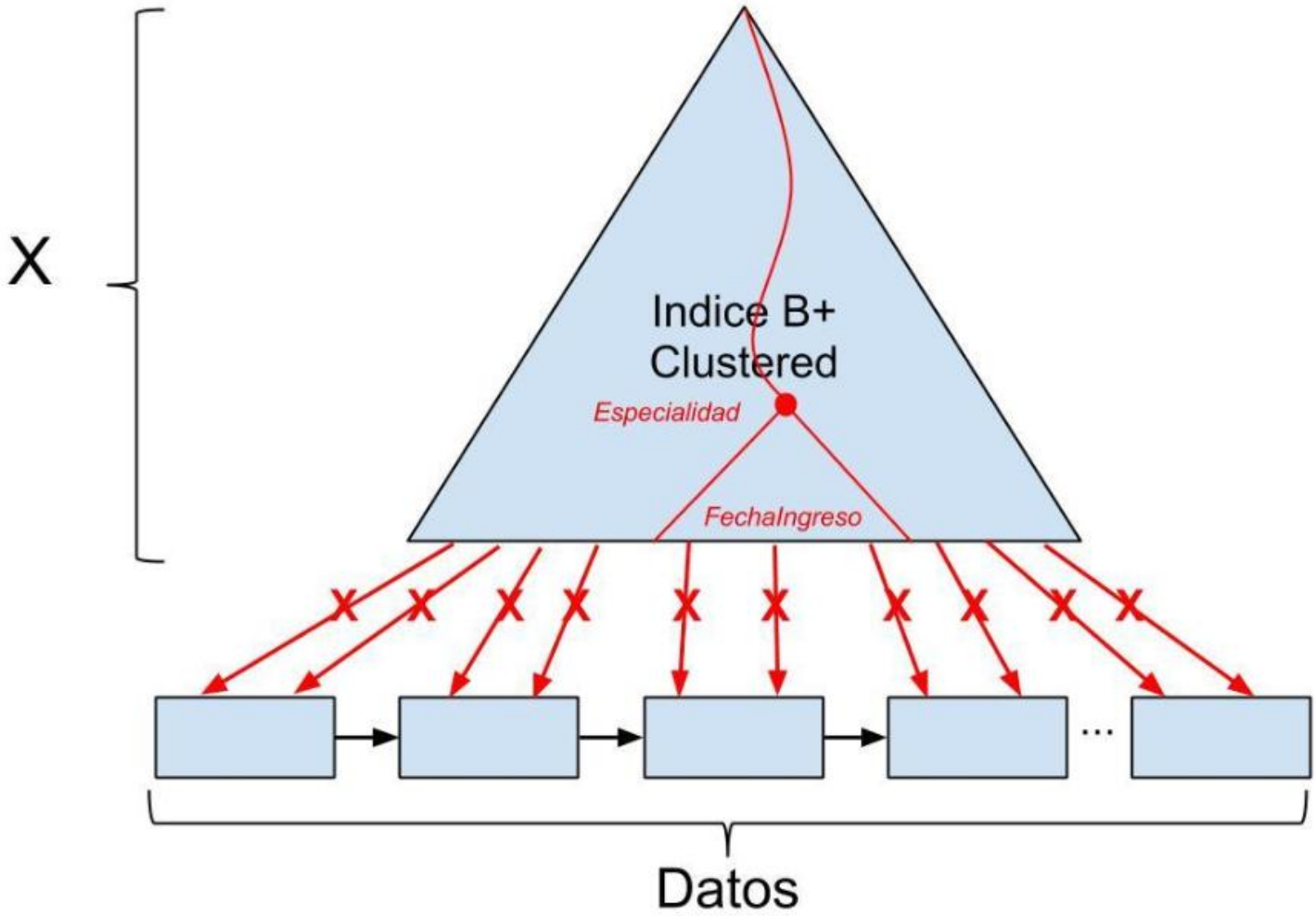
Índice

Árbol B^+ clustered sobre los atributos: <**Especialidad, FechaIngreso** >,
Altura del árbol: 4 bloques

Atención: el índice está físicamente ordenado por especialidad, y adentro de ese atributo por fechaIngreso.

Escenario 5: cuando todo está en el índice (cont.)

Gráficamente:



Escenario 5: cuando todo está en el índice (cont.)

Pasos a seguir:

Escenario 5: cuando todo está en el índice (cont.)

Pasos a seguir:

- leemos 4 bloques (la altura del árbol)

Escenario 5: cuando todo está en el índice (cont.)

Pasos a seguir:

- leemos 4 bloques (la altura del árbol)
- luego debemos leer N hojas (cantidad de fechas ingreso que cumplen con especialidad 'traumatología'). Esto puede aumentarnos la cantidad de bloques en memoria. Por simplicidad para los cálculos, asumimos que se lee 1 hoja.

Escenario 5: cuando todo está en el índice (cont.)

Pasos a seguir:

- leemos 4 bloques (la altura del árbol)
- luego debemos leer N hojas (cantidad de fechas ingreso que cumplen con especialidad 'traumatología'). Esto puede aumentarnos la cantidad de bloques en memoria. Por simplicidad para los cálculos, asumimos que se lee 1 hoja.
- **costo total: 4**