IIC2413

AYUDANTÍA 10

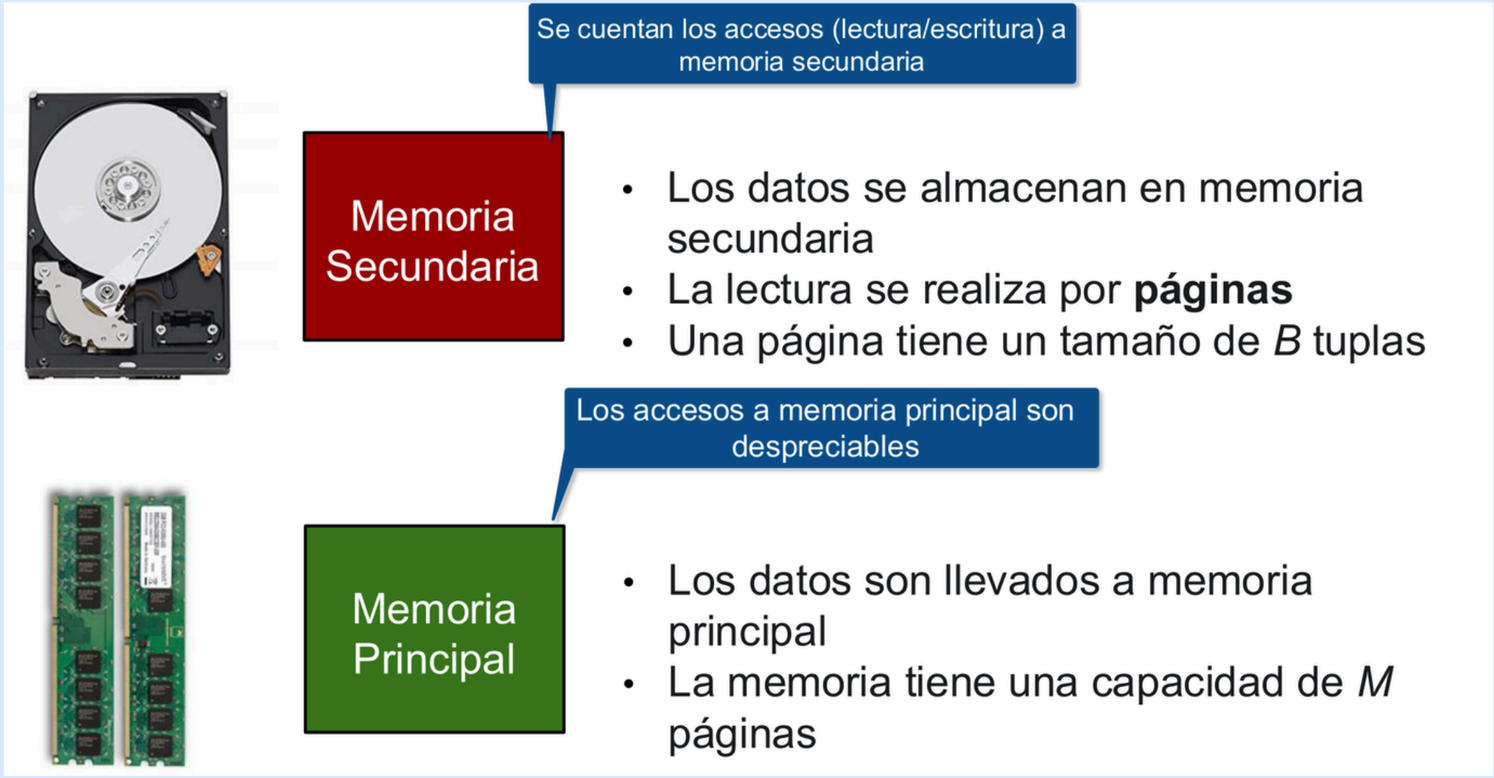
Índices

Introducción: El problema del costo

Los datos se guardan en disco y el DBMS los lleva a RAM para procesarlos.

Leer desde disco es mucho más lento que acceder a RAM.

Por eso contamos accesos al disco y despreciamos el costo en RAM.



Costo de una consulta = número de lecturas y escrituras de páginas (I/O)

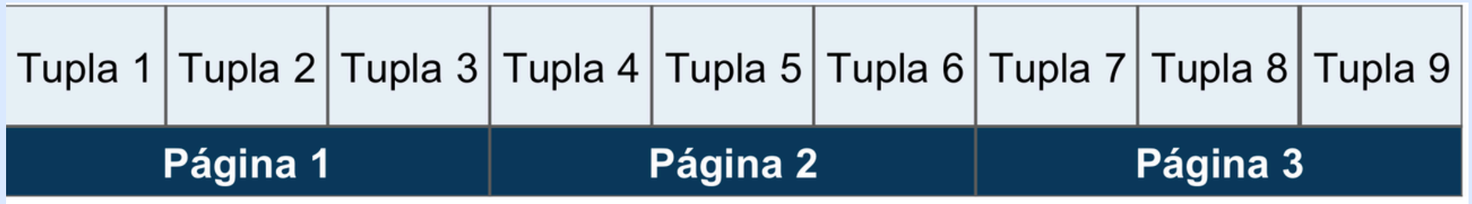
BLOQUES, PÁGINAS Y RAM

El disco almacena los datos en bloques o páginas: unidades que contienen varias tuplas.

Al leer, se copia una página completa a RAM, aunque necesitemos una sola tupla.

Ejemplo: 9 tuplas, agrupadas de 3 en 3, ocupan 3 páginas.

En la RAM se conservan las páginas que el DBMS está usando.



|R| = tuplas de la tabla B = tuplas por página M = páginas que caben en RAM

SEQUENTIAL SCAN Y CÁLCULO DE I/O

Un sequential scan recorre todas las páginas de una tabla.

Ejemplo propio: Pedidos tiene 23 tuplas. Caben 5 por página.

SELECT * FROM Pedidos

Se leen 4 páginas llenas y una quinta con 3 tuplas: 5 I/O.

La última página también cuenta, aunque no esté llena. Se supone que no estaban en RAM.

Para bases de datos gigantes este costo es altísimo.

Ejemplo: tabla Pedidos, 23 tuplas

Página 1	1	2	3	4	5
Página 2	6	7	8	9	10
Página 3	11	12	13	14	15
Página 4	16	17	18	19	20
Página 5	21	22	23	libre	libre

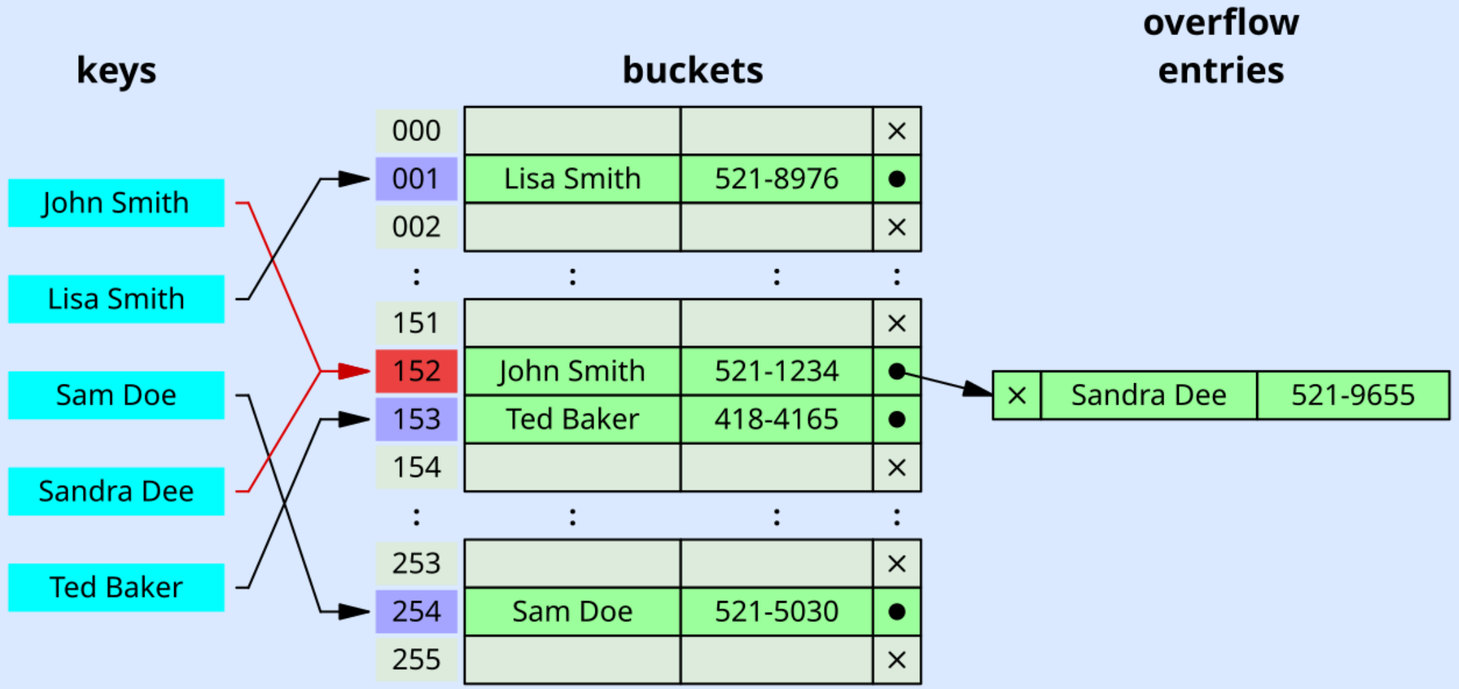
$$\text{Costo} = \lceil |R| \,/\, B \rceil = \lceil 23 \,/\, 5 \rceil = 5 \text{ I/O}$$

EDDS: FUNCIONAMIENTO DE HASH

Una función transforma la llave en un número de casillero.

Ejemplo: buscar a Sandra Dee.
1. Calcular h(Sandra Dee): 152.
2. Ir al casillero 152.
3. Revisar John Smith y seguir el enlace hasta Sandra Dee.

Ambos nombres llegan al 152: hay una colisión y se encadenan.



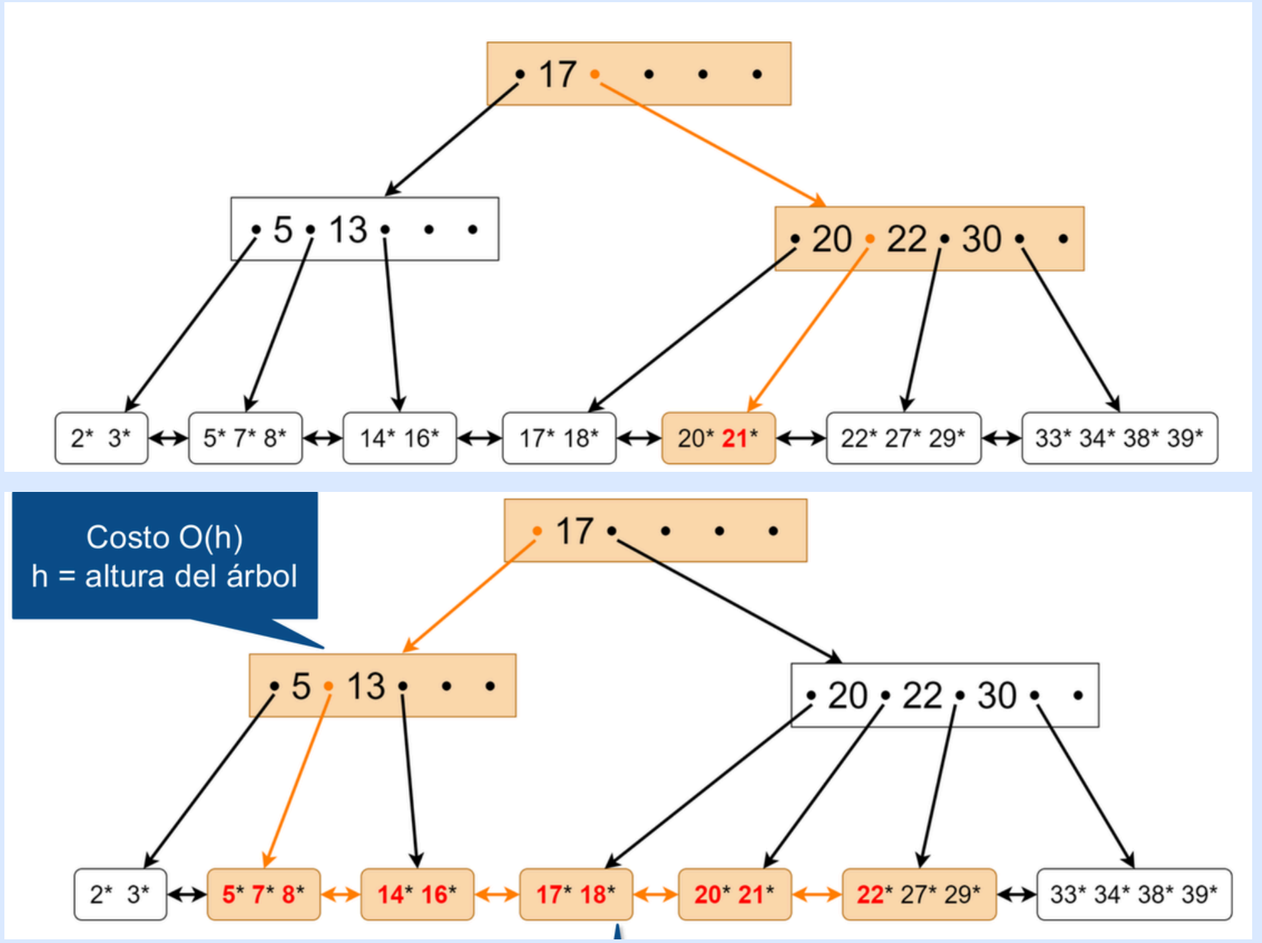
h(John Smith) = h(Sandra Dee) = 152

EDDS: FUNCIONAMIENTO DE B+

Cada nodo ocupa una página y contiene llaves que separan valores.

BUSCAR 21 (arriba)
Comparar con 17, seguir al nodo 20 / 22 / 30 y llegar a la hoja 20 / 21.

BUSCAR DE 5 A 26 (abajo)
Llegar a la hoja de 5 y seguir las hojas hasta superar 26.



Los nodos guían la búsqueda; las hojas ordenadas y enlazadas permiten recorrer rangos

EDDS: PUNTEROS

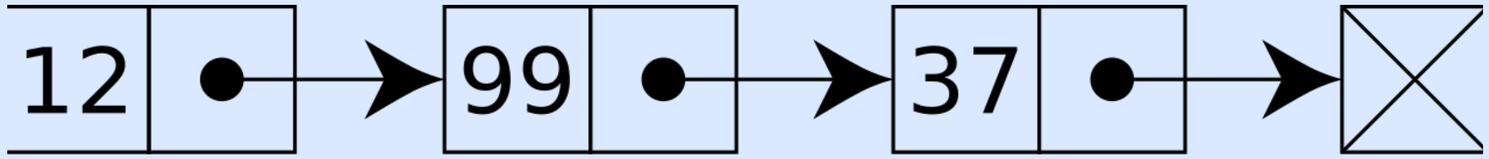
Cada variable tiene una posición en memoria, un tamaño y un valor.

Un puntero guarda la dirección donde está otro dato.

Los nodos no necesitan estar juntos en RAM. NULL marca el final.

En un índice, la referencia permite llegar a la página o registro.

Ejemplo: el nodo 12 está en la dirección 100 y apunta a 340, donde está 99. Este apunta a 220, donde está 37.



Dirección	Valor	Puntero siguiente
100	12	340
340	99	220
220	37	NULL

Un puntero guarda una dirección: permite llegar a otro dato o nodo

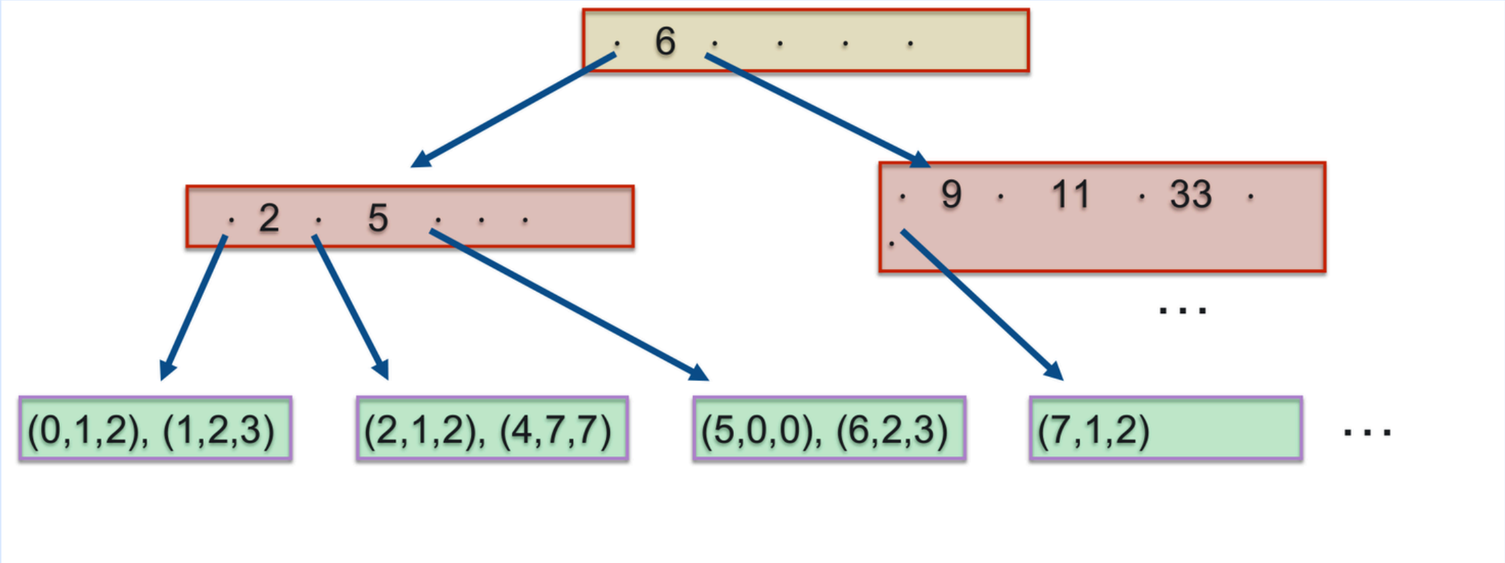
ÍNDICES: QUÉ SON Y PARA QUÉ SIRVEN

Un índice es una estructura de acceso que permite localizar tuplas sin recorrer siempre toda la tabla.

La search key es el valor buscado. La data entry contiene la tupla o una referencia a ella.

Ejemplo del dibujo: buscar a = 5. El índice guía hasta la hoja que contiene la tupla (5, 0, 0).

Sirve para acelerar consultas por igualdad o por rango, según su estructura.



Llave de búsqueda → entrada del índice → tupla buscada

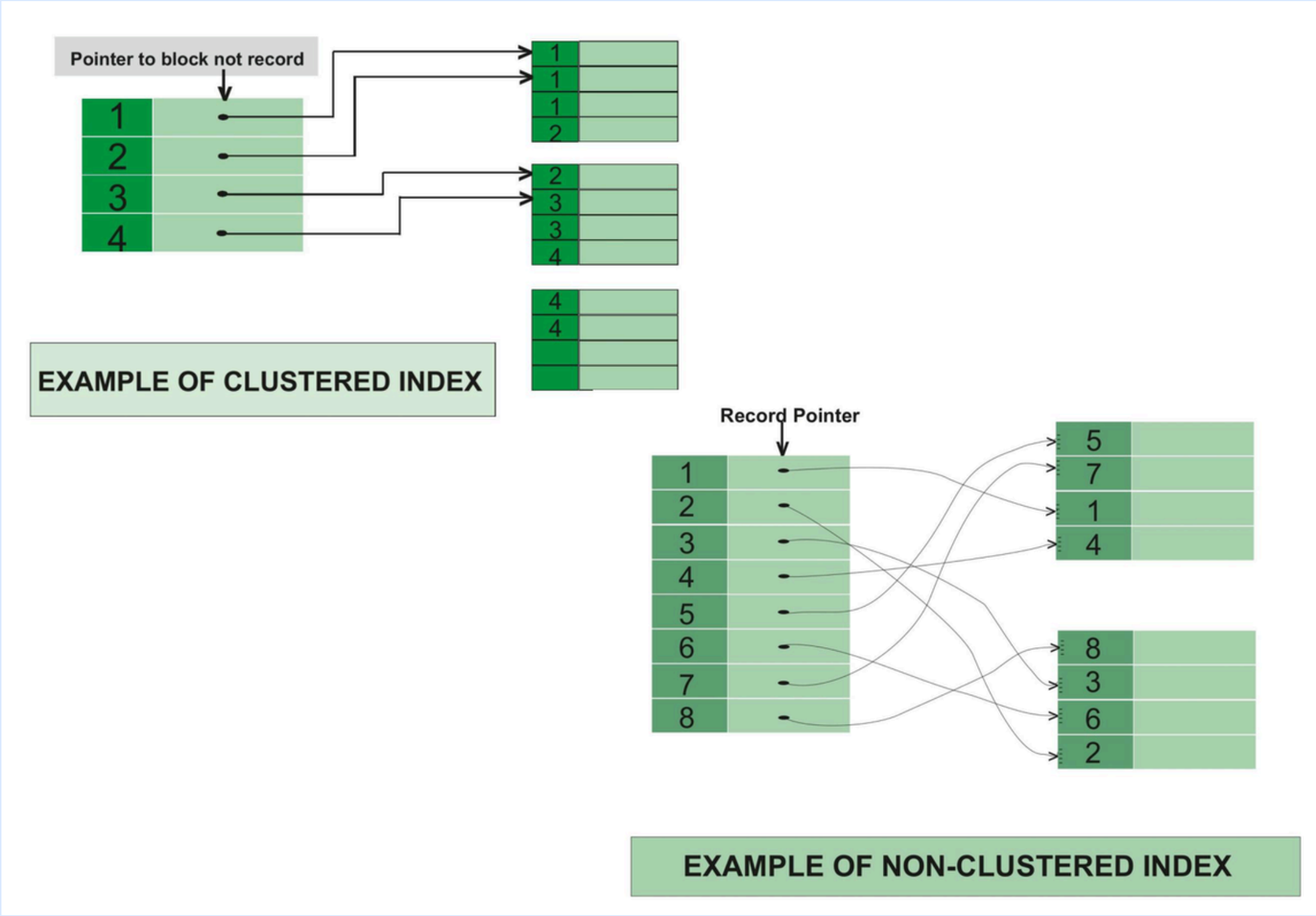
CLUSTERED VS UNCLUSTERED

CLUSTERED
Los datos siguen la organización del índice. Las tuplas próximas se concentran en menos páginas.

UNCLUSTERED
Las entradas apuntan a tuplas que pueden estar dispersas en distintas páginas.

Clustered guarda tuplas;
Unclustered guarda punteros.

La diferencia importa al recuperar muchos resultados.



Datos agrupados: menos páginas para recuperar resultados próximos

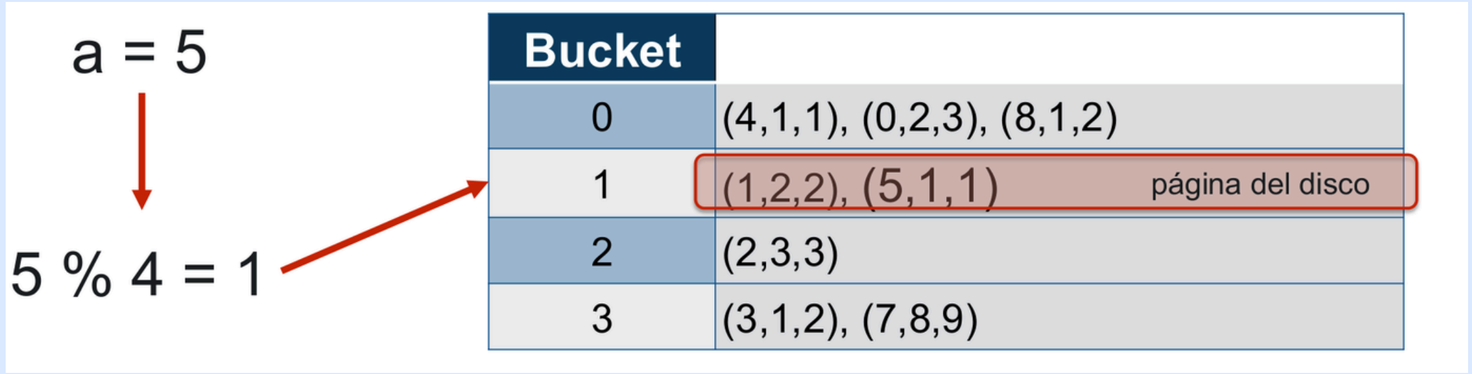
HASH INDEX Y FÓRMULA DE COSTO

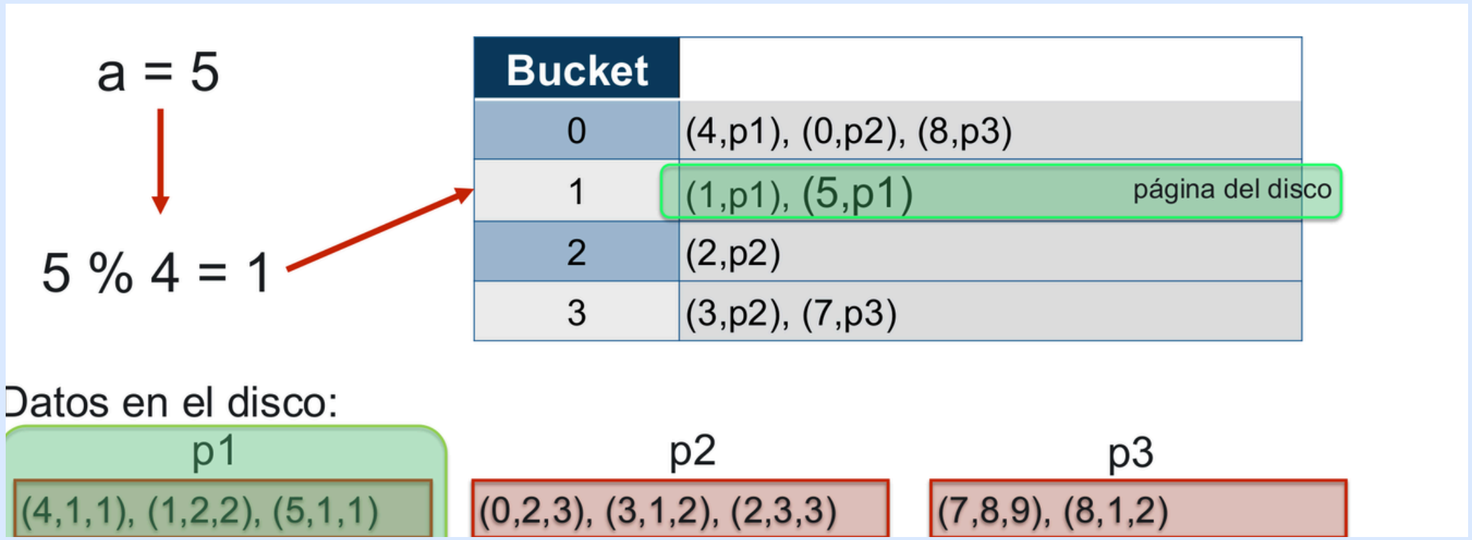
La función hash indica qué bucket leer. En disco, un bucket contiene una página de entradas.

Ejemplo: a = 5.
5 mod 4 = 1: leer el bucket 1.

CLUSTERED
La tupla ya está en ese bucket.

UNCLUSTERED
La entrada (5, p1) obliga a leer también la página p1.





Igualdad: clustered = 1 I/O unclustered = 1 + 1 = 2 I/O

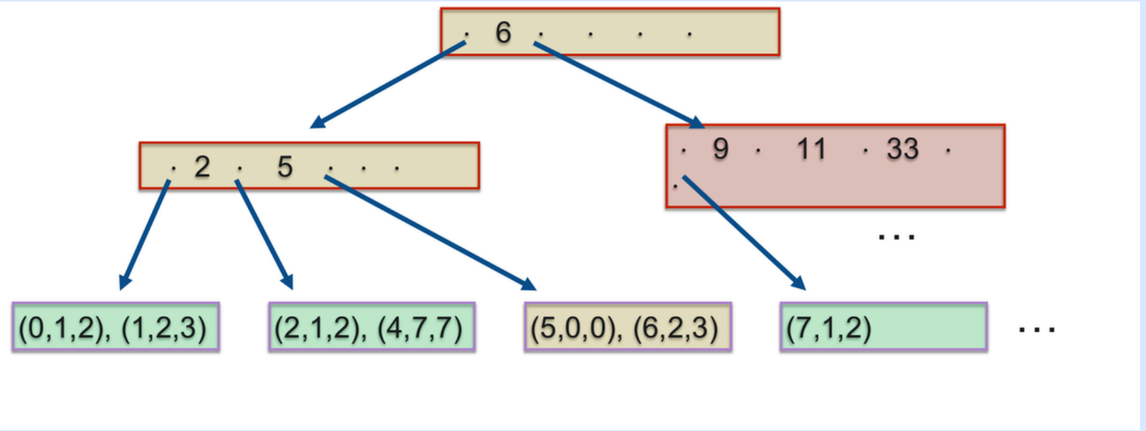
B+TREE INDEX Y FÓRMULA DE COSTO

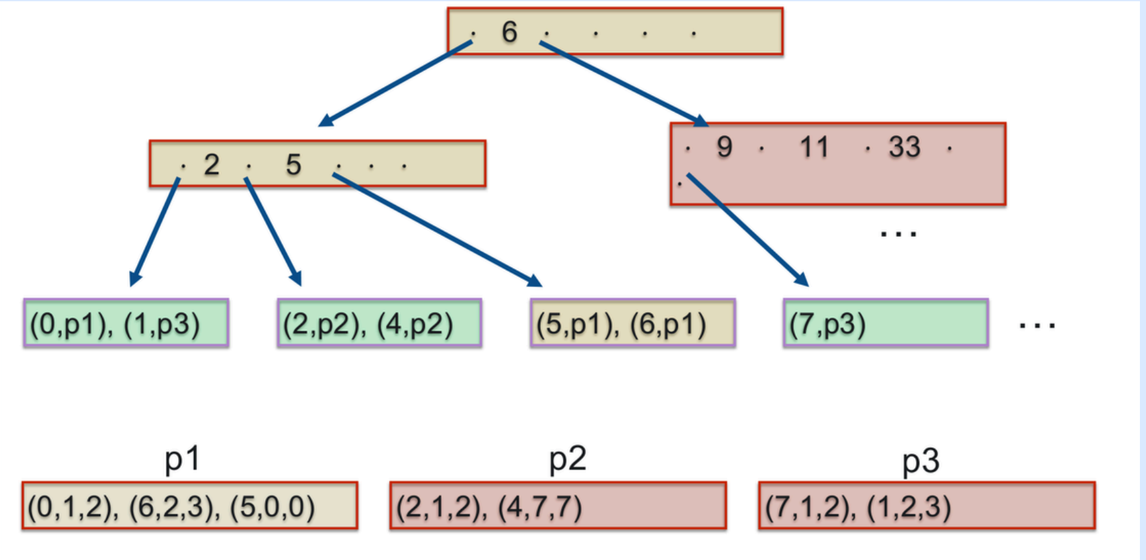
h cuenta las páginas desde la raíz hasta la hoja, incluida. (altura)

Ejemplo: a = 5.
Se sigue el camino hasta la hoja con la entrada 5.

CLUSTERED
La hoja contiene la tupla: h lecturas.

UNCLUSTERED
La hoja contiene (5, p1): se añade una lectura de p1.





Igualdad: clustered = h I/O unclustered = h + 1 I/O

TABLA COMPARATIVA: VENTAJAS Y DESVENTAJAS

Hash / B+: estructura del índice. Clustered / Unclustered: organización de los datos.

OPCIÓN	VENTAJA	DESVENTAJA / LÍMITE
Hash	Igualdad rápida: costo esperado O(1).	No ordena las llaves. Los rangos pueden exigir un scan.
Árbol B+	Sirve para igualdad y rangos. Hojas ordenadas y enlazadas.	Hay que recorrer niveles del árbol y luego recuperar los resultados.
Clustered	Las tuplas próximas en el índice se recuperan en pocos bloques.	La organización de los datos favorece ese criterio de acceso.
Unclustered	Permite otro acceso mediante punteros sin reorganizar los datos por esa llave.	Las tuplas pueden estar dispersas: más lecturas para muchos resultados.

EJERCICIOS

EJERCICIOS

Dada la siguiente tabla de la base de datos de la universidad, responda las siguientes preguntas:

```
CREATE TABLE Estudiantes (
    n_alumno SERIAL PRIMARY KEY,          -- Número de alumno único por carrera
    nombre VARCHAR(100) NOT NULL,         -- Nombre del estudiante
    RUN VARCHAR(10) NOT NULL,             -- Rol Único Nacional (formato NNNNNNNN-A)
    carrera VARCHAR(50) NOT NULL,         -- Nombre de la carrera
    PPA DECIMAL(3,2) NOT NULL             -- Promedio ponderado acumulado
);
```

EJERCICIOS

a) ¿Qué ventaja tendría para el negocio crear un índice sobre la columna `RUN`? Explique. (3 pts)

EJERCICIOS

a) ¿Qué ventaja tendría para el negocio crear un índice sobre la columna `RUN`? Explique. (3 pts)

Solución: Es muy usual que los estudiantes se sepan su RUN y menos que se sepan el número de alumno, por lo que agregar este índice secundario da un segundo mecanismo de búsqueda eficiente, si no se agrega un índice a RUN la búsqueda necesita un full scan.

EJERCICIOS

c) ¿Para qué tipo de búsqueda, en esta tabla, se recomienda el método B+ y Hash? (4 pts)

EJERCICIOS

c) ¿Para qué tipo de búsqueda, en esta tabla, se recomienda el método B+ y Hash? (4 pts)

Solución: El método B+ se recomienda para búsquedas de rango y el método Hash para búsquedas exactas.

EJERCICIOS

d) ¿Qué impacto tiene en el rendimiento de las consultas el uso excesivo de índices en una tabla? (4 pts)

EJERCICIOS

d) ¿Qué impacto tiene en el rendimiento de las consultas el uso excesivo de índices en una tabla? (4 pts)

Solución: El uso excesivo de índices puede degradar el rendimiento de las operaciones de escritura (INSERT, UPDATE, DELETE), ya que cada índice debe actualizarse. También aumenta el uso de almacenamiento y puede hacer más compleja la administración de la base de datos.

EJERCICIOS

e) **[Bonus]** Si los índices primarios de la tabla se implementan usando B+ Clustered y realizamos una búsqueda por el campo `n_alumno=12345678`, ¿cuál es el número de I/O de la búsqueda? (Hint: suponga árbol de altura h y que cada página contiene P tuplas). (3 pts)

EJERCICIOS

e) **[Bonus]** Si los índices primarios de la tabla se implementan usando B+ Clustered y realizamos una búsqueda por el campo `n_alumno=12345678`, ¿cuál es el número de I/O de la búsqueda? (Hint: suponga árbol de altura h y que cada página contiene P tuplas). (3 pts)

Solución: En B+ Clustered los datos se encuentran en las hojas del árbol. Cómo en este caso se busca un valor específico, solo es necesario navegar a través del árbol hasta la hoja y en ese caso el costo de I/O es la altura del árbol h. Si el índice está cargado en memoria el costo de I/O es despreciable.

EJERCICIOS

e) **[Bonus]** Si los índices primarios de la tabla se implementan usando B+ Clustered y realizamos una búsqueda por el campo `n_alumno=12345678`, ¿cuál es el número de I/O de la búsqueda? (Hint: suponga árbol de altura h y que cada página contiene P tuplas). (3 pts)

Y SI SE TIENE B+ UNCLUSTERED?