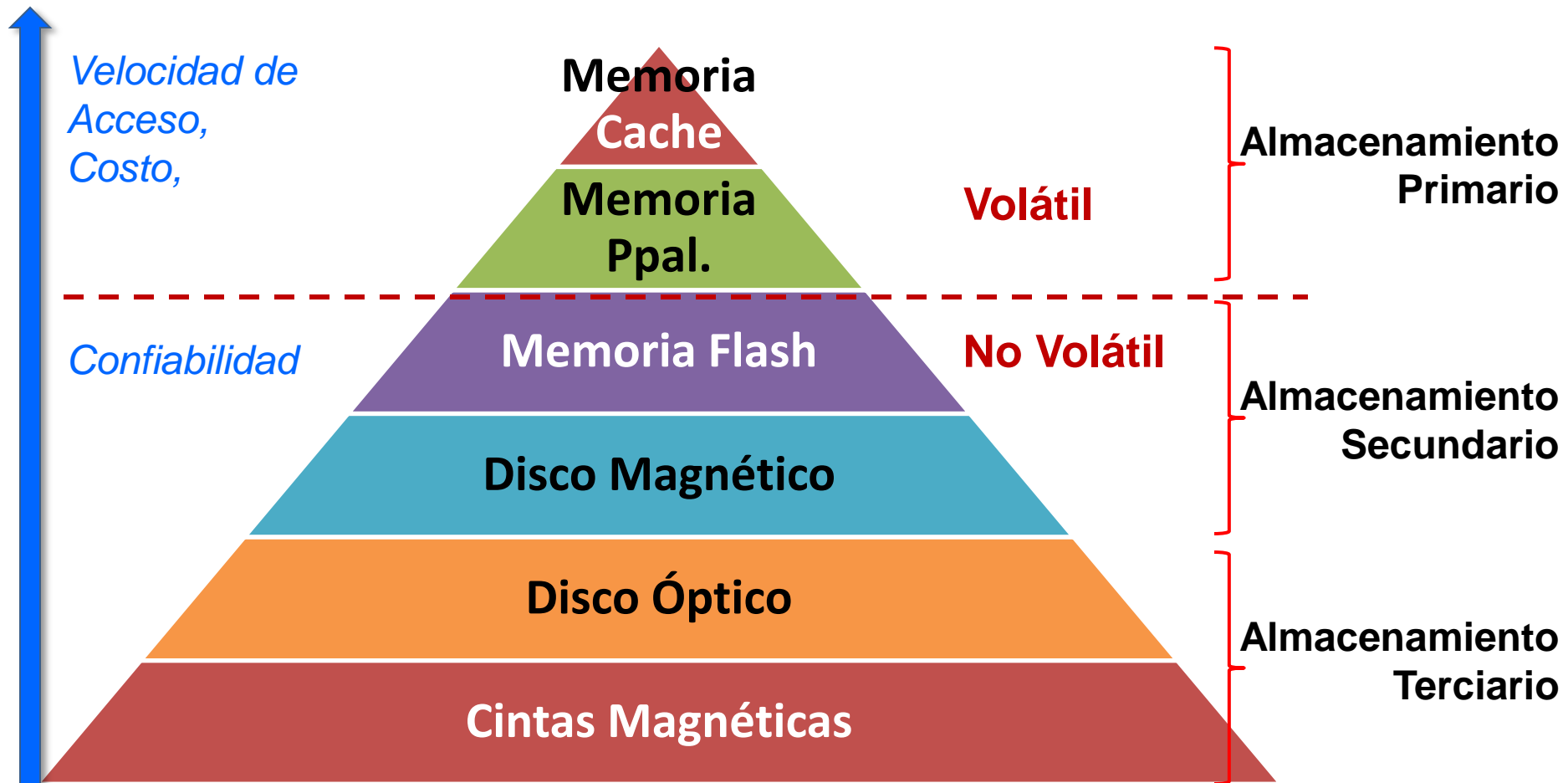
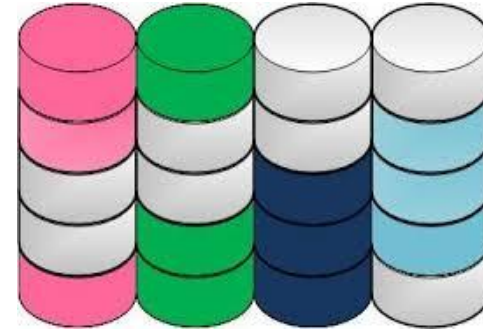


1) Explicar la organización de los medios de almacenamiento en primarios, secundarios y terciarios.

Jerarquía de Almacenamiento



2) ¿Qué aspectos se deben considerar al analizar un dispositivo de almacenamiento físico?



- Capacidad
 - Velocidad de Acceso
 - Costo por unidad
 - Confiabilidad
- **Tiempo de Acceso:** tiempo transcurrido desde un pedido de lectura/escritura hasta que comienza la transferencia de datos.
 - *Tiempo de búsqueda.*
 - *Tiempo de latencia por rotación.*
 - **Tiempo de Transferencia:** velocidad a la que se recuperan o guardar datos en el disco.
 - **Tiempo medio entre fallos:** tiempo promedio que se puede esperar que funcione sin fallos.

3) Definir registro y bloque. Proponga una organización simple de una base para una bases de datos relacional

Una organización simple sería:

- **Un archivo por relación.**
- Todos los registros de un archivo son de igual tamaño (registros de tamaño fijo) y tienen los mismos atributos.
- **Acomodar secuencialmente registros completos en cada bloque.**

4 a) Explicar el proceso de I/O a través del buffer de memoria.

- Por *cada solicitud de un bloque de dato b_i*
 1. Si b_i está en el buffer de memoria, BM retorna la dirección de b_i en memoria principal.
 2. Si b_i no está en el buffer de memoria, BM debe:
 - i. **Asignar espacio** del buffer para el nuevo bloque:
Remplazando algún otro bloque, si es necesario hacer espacio, y pero escribiendo previamente el bloque remplazado en disco si es que fue modificado desde la última vez que se copio en disco.
 - ii. **Leer el bloque desde el disco para copiarlo en el buffer**, y retornar la dirección de memoria principal que aloja el bloque.

6 a) Qué objetivos persigue una organización de discos RAID.

RAIDs - Redundant Array of Independent Disks - Técnica para organizar y administrar varios discos físicos como si fueran una unidad

- + Aumenta la *capacidad*
- + Aumenta la *velocidad de acceso* a la información usando múltiples discos en paralelo.
- + Aumenta la *confiabilidad*, dando la oportunidad de almacenar datos redundantes, de forma que pueda recuperarse información, aún ante la presencia de fallos.

6 b y c) Como funciona RAID 1 y 5.

- **RAID Level 0:** Block striping; non-redundant.
- **RAID Level 1:** Mirrored disks with block striping.
- **RAID Level 0+1 o 10**
- **RAID Level 2:** (ECC) with bit striping.
- **RAID Level 3:** Bit-Interleaved Parity.
- **RAID Level 4:** Block-Interleaved Parity.
- **RAID Level 5:** Block-Interleaved Distributed Parity.
- **RAID Level 6:** P+Q Redundancy.



RAID 1: mirrored disks



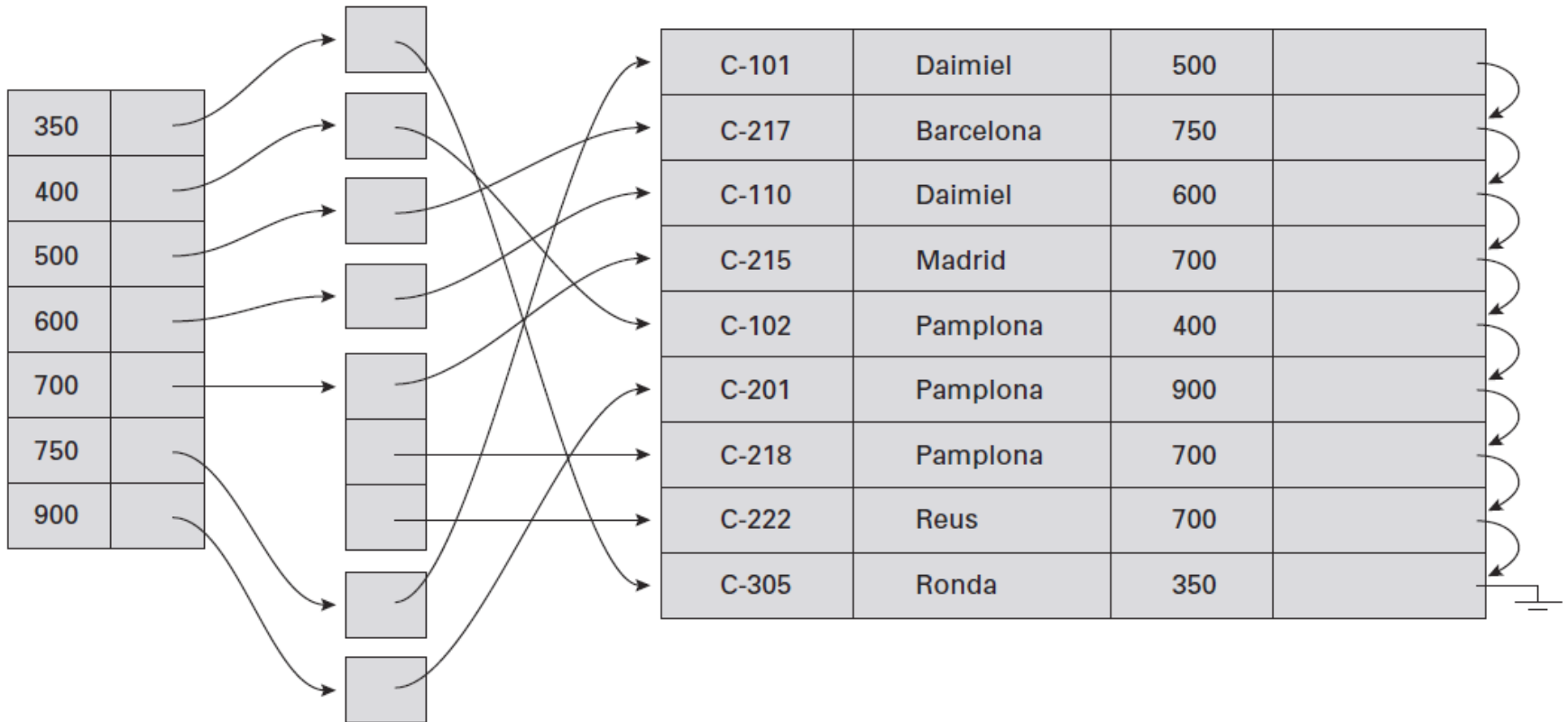
RAID 5: block-interleaved distributed parity

7 a) ¿Qué diferencia existe entre índice primario y secundario?

- El índice se asocia con una *clave_de_búsqueda*. El **archivo índice** almacena de manera ordenada los valores de las claves de *búsqueda*, y asocia con cada clave los registros de datos que contienen esa clave de búsqueda.
- Si el **archivo de datos está ordenado secuencialmente**, el índice cuya clave de búsqueda especifica el orden secuencial del archivo es el **índice primario**.
 - El índice primario NO necesita ser la clave primaria.
 - Los índices que tienen el orden secuencial de los registros de datos se denominan **índices clustering**
- Los índices cuyas claves de búsqueda especifican un orden diferente del orden secuencial del archivo se llaman **índices secundarios o no clustering**.

7 a) ¿Qué diferencia existe entre índice primario y secundario?

- **índices secundarios o no clustering.**



7 b) ¿El índice primario es la llave primaria de una relación?

- Puede no serlo.

7 c) ¿Cuántos índices primarios puede tener un archivo? ¿Cuántos índices secundarios?

- Sólo puede haber un índice primario, ya que requiere que los registros estén ordenados secuencialmente. Pueden ser densos o ralos (sparse)
- Pueden haber muchos índices secundarios. Los cuales sólo pueden ser densos.

8) ¿Cuándo es preferible utilizar un índice denso en vez de un índice ralo?

- Son preferibles cuando hay muchas más **consultas** que inserciones y borrados
- El índice ralo requiere menos espacio y menos overhead por tareas ingresar/borrar registros.
- El índice ralo solo se puede usar cuando el archivo de datos ordenados como la clave del índices (índice primario o clustering).
- los índices secundarios se usan para recorrer los datos en un orden distinto al orden físico. Es por eso que deben ser densos y con una entrada en el índice por cada valor de la clave de búsqueda, y un puntero a cada registro del archivo.

Índice Denso

Biology		76766	Crick	Biology	72000	
Comp. Sci.		10101	Srinivasan	Comp. Sci.	65000	
Elec. Eng.		45565	Katz	Comp. Sci.	75000	
Finance		83821	Brandt	Comp. Sci.	92000	
History		98345	Kim	Elec. Eng.	80000	
Music		12121	Wu	Finance	90000	
Physics		76543	Singh	Finance	80000	
		32343	El Said	History	60000	
		58583	Califieri	History	62000	
		15151	Mozart	Music	40000	
		22222	Einstein	Physics	95000	
		33465	Gold	Physics	87000	

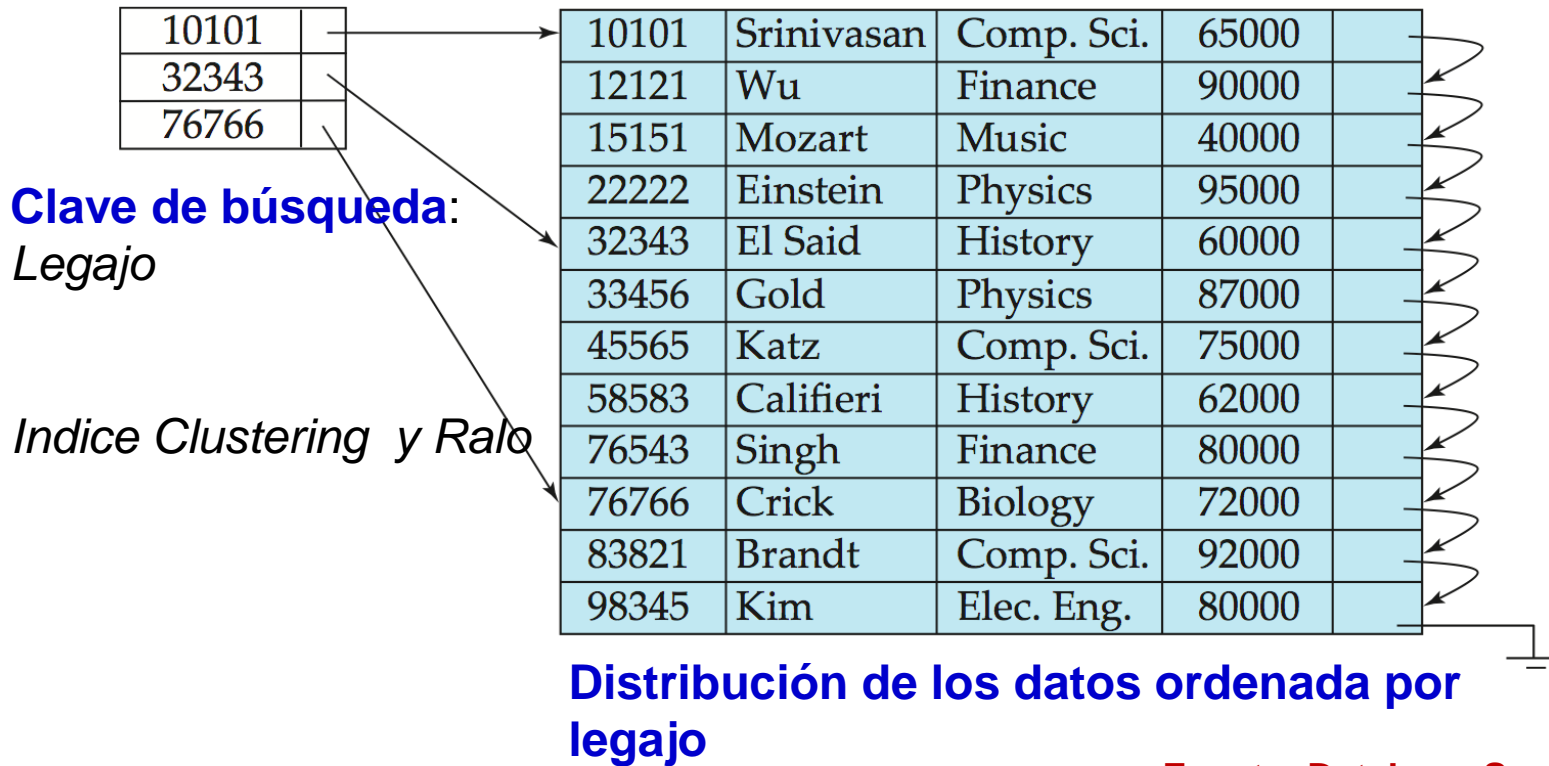
Clave de búsqueda:
departamento

Índice clustering y denso

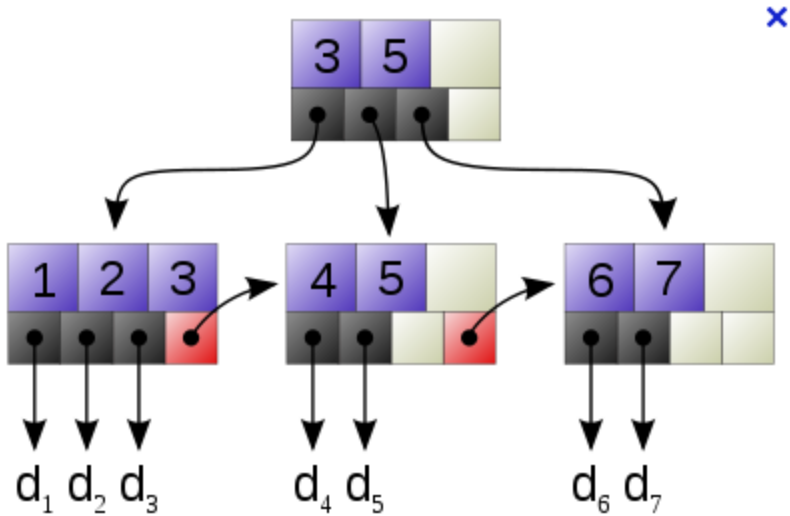
Distribución de los datos ordenada por departamento

**Fuente: Database System Concepts
Silberschatz, Korth, Sudarshan**

Índice Ralo



**Fuente: Database System Concepts
Silberschatz, Korth, Sudarshan**



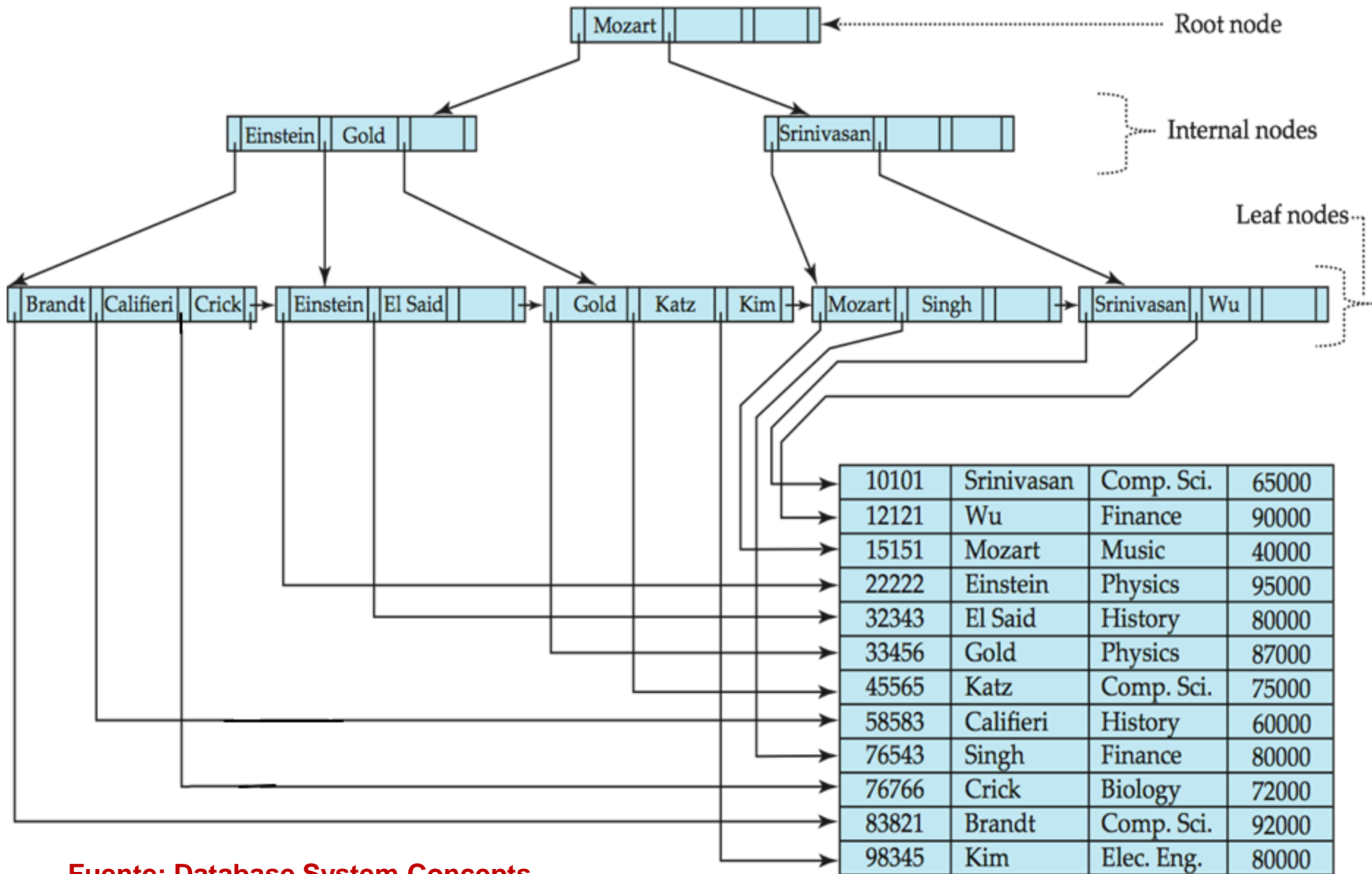
B+ Árboles

Un índice de árbol B+ es un índice multinivel con una estructura que difiere del índice multinivel de un archivo secuencial

Índices B+ Árbol

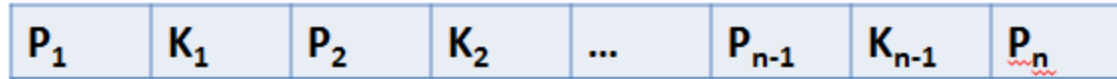
- Un índice B+ árbol es un **árbol balanceado** donde los caminos desde la raíz a cada hoja del árbol son de la misma longitud.
- Cada nodo que no es una hoja tiene entre $\lceil n/2 \rceil$ y n hijos, con n es fijo para cada árbol. El nodo raíz es un caso especial
- Un nodo típico de un árbol B+ contiene hasta $n - 1$ claves de búsqueda K_1, K_2, \dots, K_{n-1} y n punteros P_1, P_2, \dots, P_n .
 - K_i claves de búsqueda, $K_1 < K_2 \dots$
 - P_i punteros a hijos, para nodos intermedios o punteros a registros o buckets para nodos hijos

P_1	K_1	P_2	K_2	...	P_{n-1}	K_{n-1}	P_n
-------	-------	-------	-------	-----	-----------	-----------	-------



Fuente: Database System Concepts
Silberschatz, Korth, Sudarshan

Nodos del B+



- **Nodos hoja:**

- P_i apunta al registro con clave K_i .
- P_n apunta a la próxima hoja.

- **Nodos intermedios:**

- P_1 apunta al subárbol que contiene claves menores a K_1 .
- P_2, \dots, P_{n-1} apuntan al subárbol que contiene claves mayores o iguales a K_{i-1} y menores a K_i .
- P_n apunta al subárbol que contiene claves mayores o iguales a K_{n-1}

B+ Árboles

Dado un B+ Árbol con $n = 6$

- Los **nodos hoja** tienen entre $\lceil (n-1)/2 \rceil$ and $n-1$, valores (entre 3 y 5 valores para $n = 6$).
- Los **nodos intermedios** (no raíz) deben tener $\lceil n/2 \rceil$ and n hijos (con entre 3 y 6 hijos $n = 6$).
- La **raíz** debe tener por lo menos dos hijos
- Una característica de los índices B+ es que se mantienen **balanceados**. Esto es, el camino desde la raíz a cada nodo hoja es la misma. Esto asegura un buen rendimiento para las búsquedas, inserciones y borrados.

Consultas con B+ Árbol

- Si existen K valores para clave de búsqueda en el archivo, el camino no será más largo que $\log_{n/2} (K)$
- Generalmente, cada nodo tiene el mismo tamaño que un bloque de disco.
- Para una **clave de búsqueda de 12 bytes** y un **tamaño del puntero a disco de 8 bytes**, n está alrededor de 200. Una estimación más conservadora n está en torno a 100.
 - Con $n = 100$, si se tienen un millón de valores de clave de búsqueda en el archivo, una búsqueda necesita solamente 4 accesos a nodos.
- Normalmente, el nodo raíz del árbol se guarda en memoria intermedia; así se necesitan tres o menos lecturas del disco.

Actualizaciones con B+ Árbol

- El borrado y la inserción son más complejos, ya que podría ser necesario **dividir** un nodo que resultara demasiado grande como resultado de una inserción, o **fusionar** nodos si un nodo se volviera demasiado pequeño.
- También se debe asegurar que el equilibrio del árbol se mantiene, con lo cual la operación se puede expandir a otros niveles.