

# Organización física de archivos e índices

Sivana Hamer - [sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)  
Escuela de Ciencias de la Computación  
Licencia: CC BY-NC-SA 4.0

# Podemos guardar datos de distintas maneras



## Memoria primaria

- Volatil
- Rapido
- Caro
- Poca capacidad alm.



## Memoria secundaria

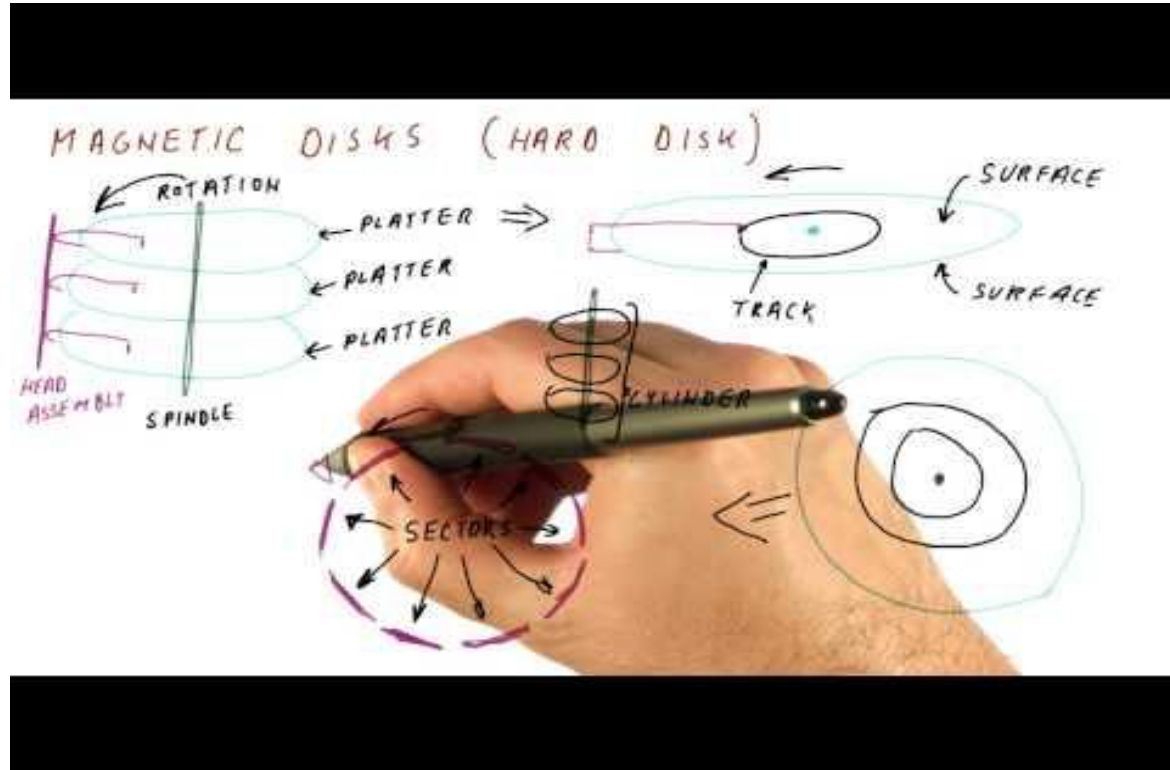
- No volatil
- Lenta
- Barato
- Mayor capacidad alm.



## Memoria ternaria

- No volatil
- Similar a secundaria
- Medio desmontable
- Offline

Las bases de datos guardan los datos en memoria no volátil  
(en memoria magnetica)

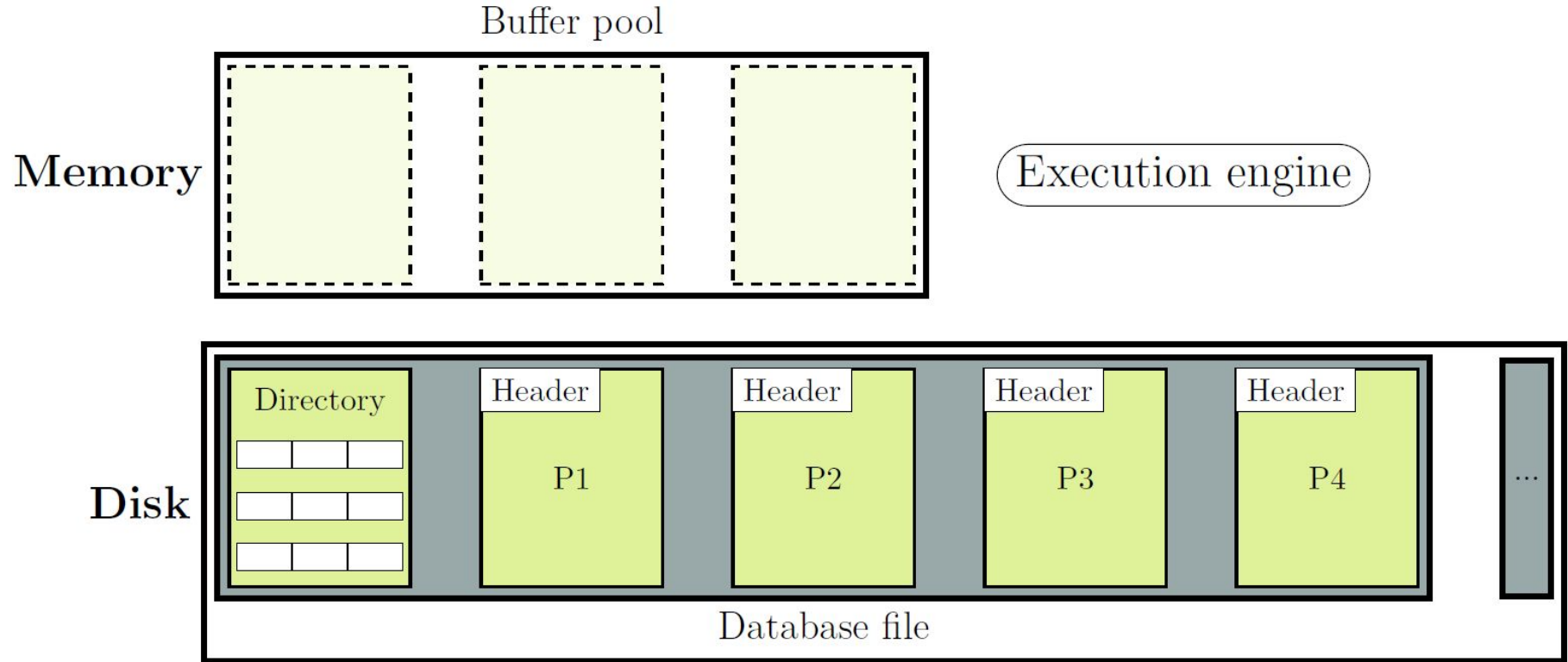


# Se guardan respaldos de los datos en cintas magnéticas

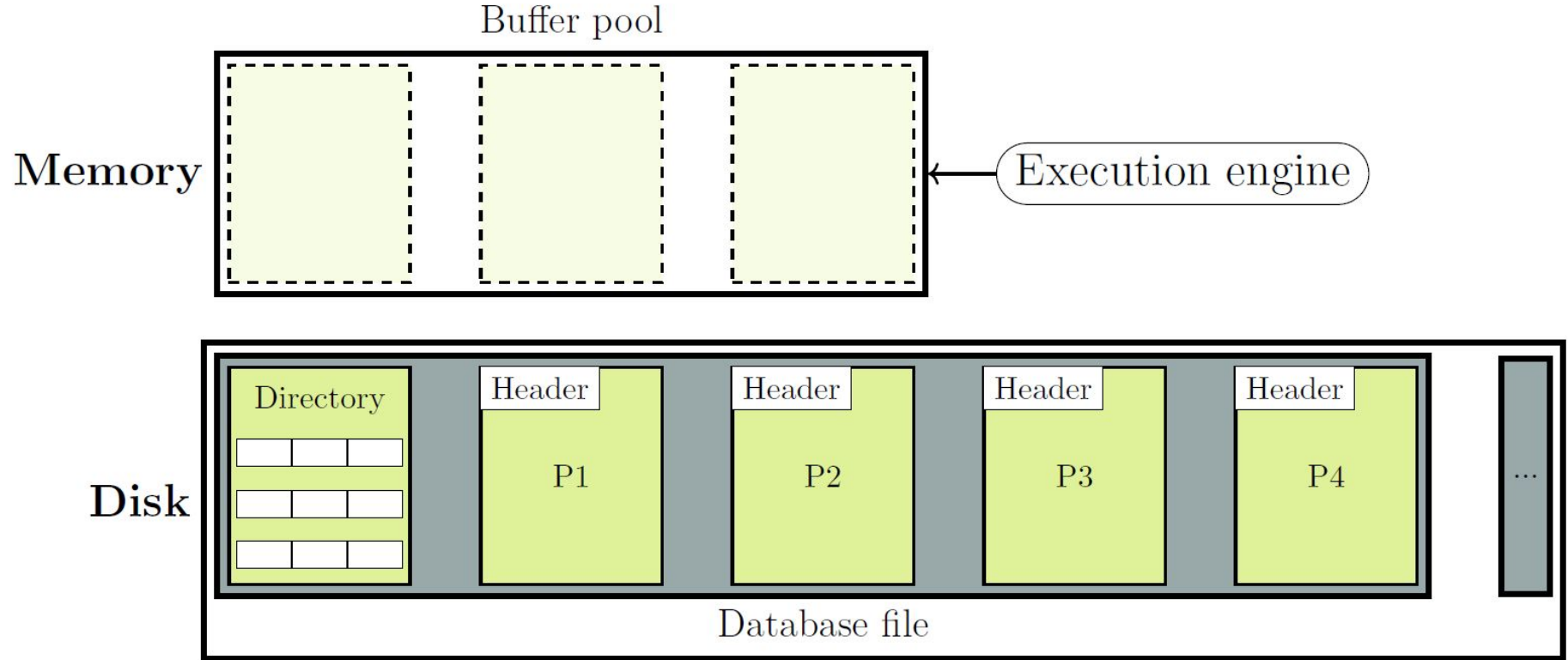




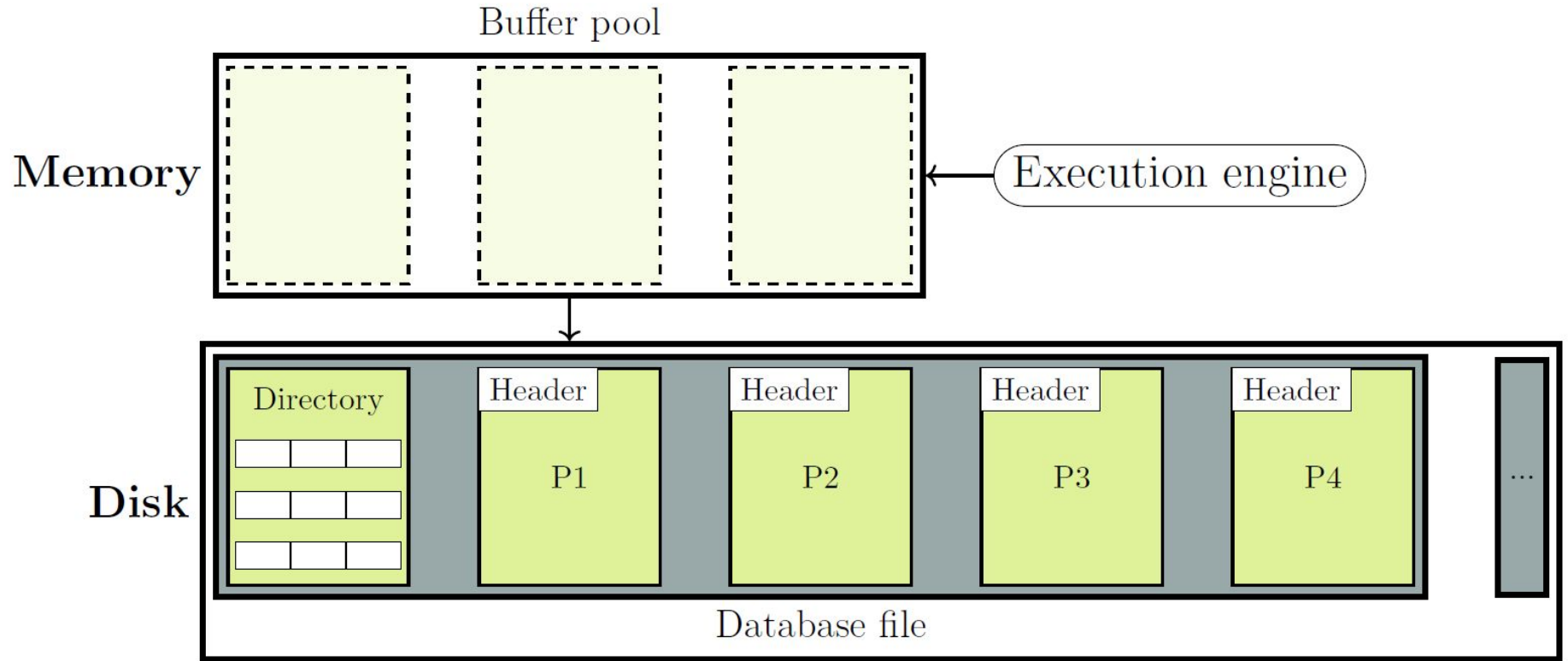
El diseño físico de una BD *disk-oriented* es...



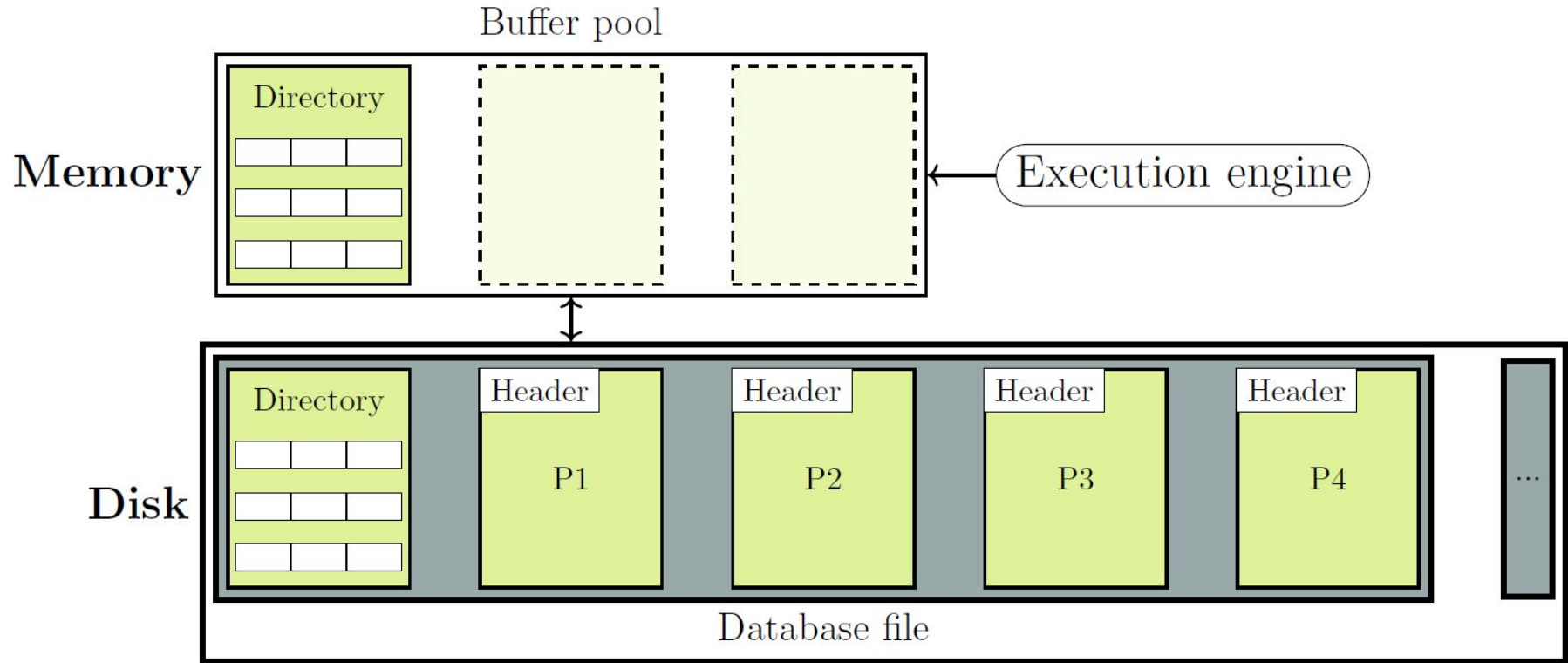
El diseño físico de una BD *disk-oriented* es...



El diseño físico de una BD *disk-oriented* es...

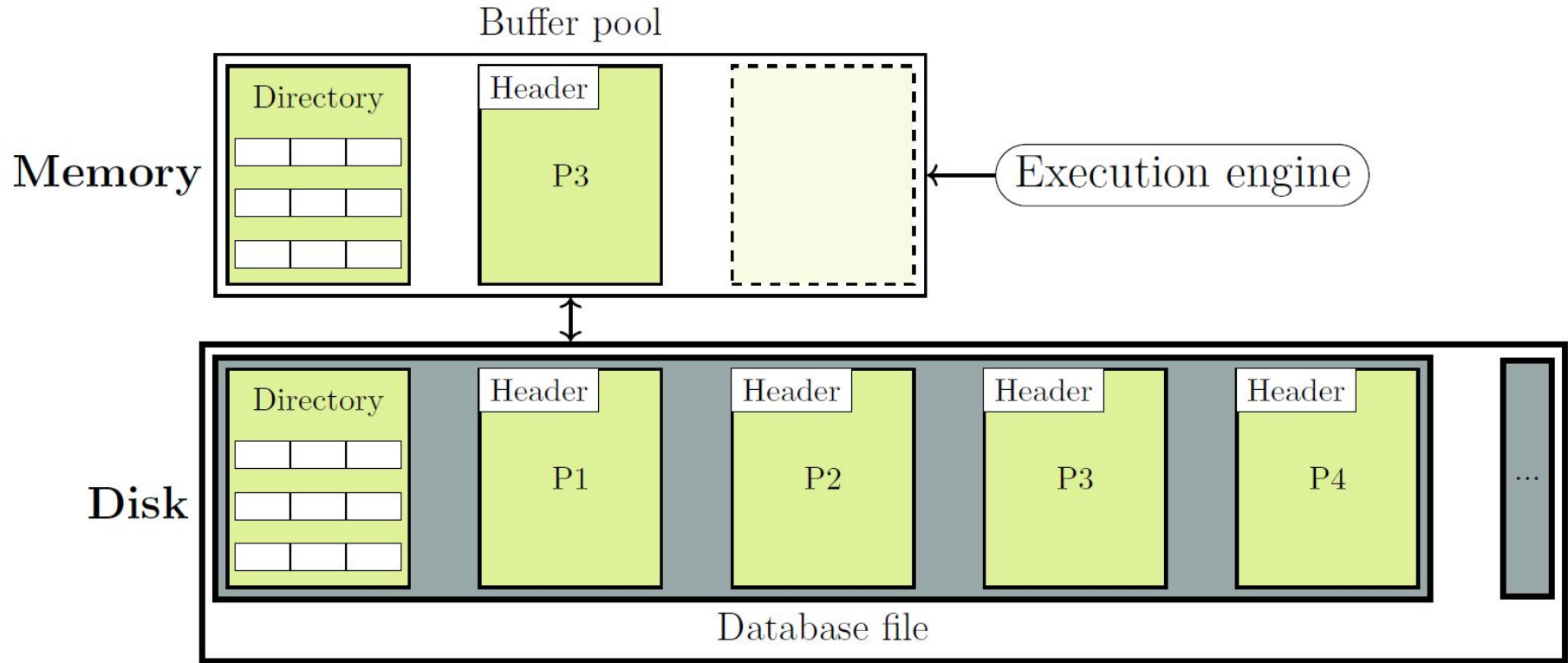


El diseño físico de una BD *disk-oriented* es...

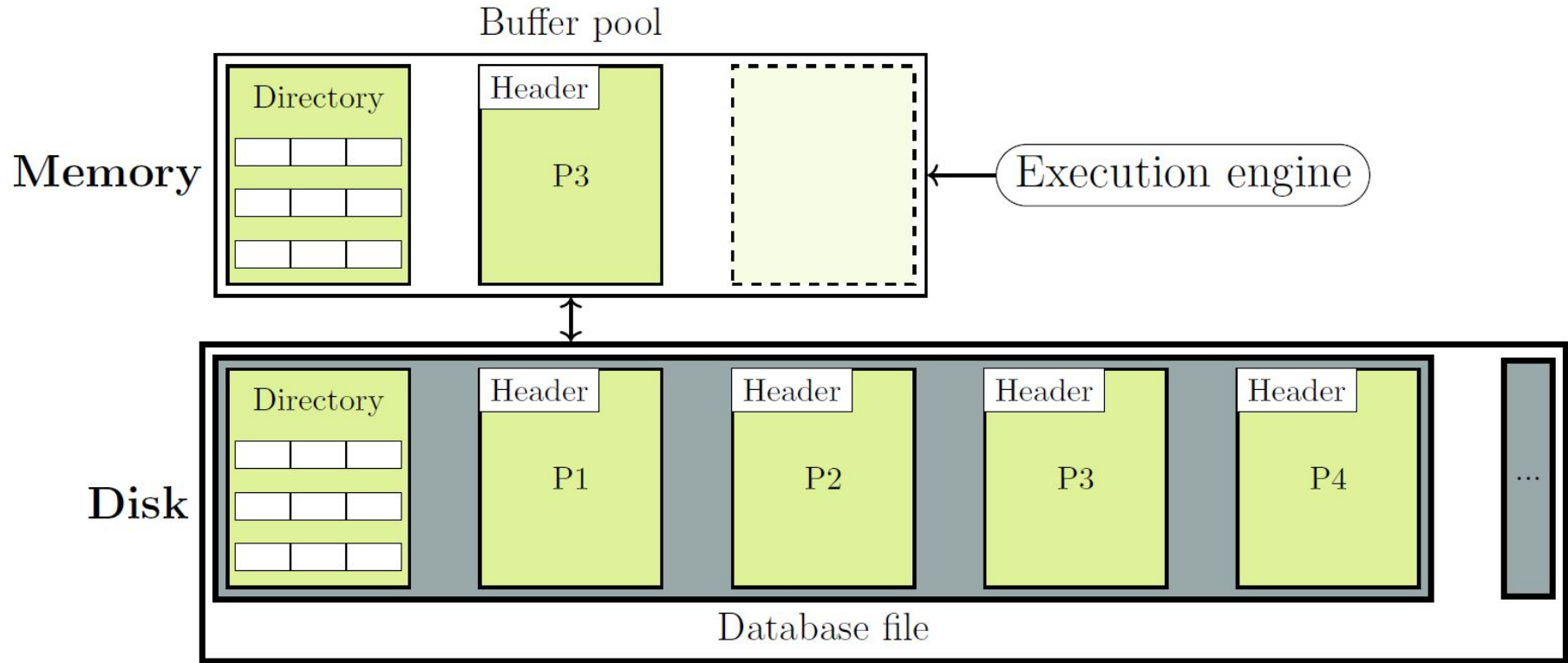




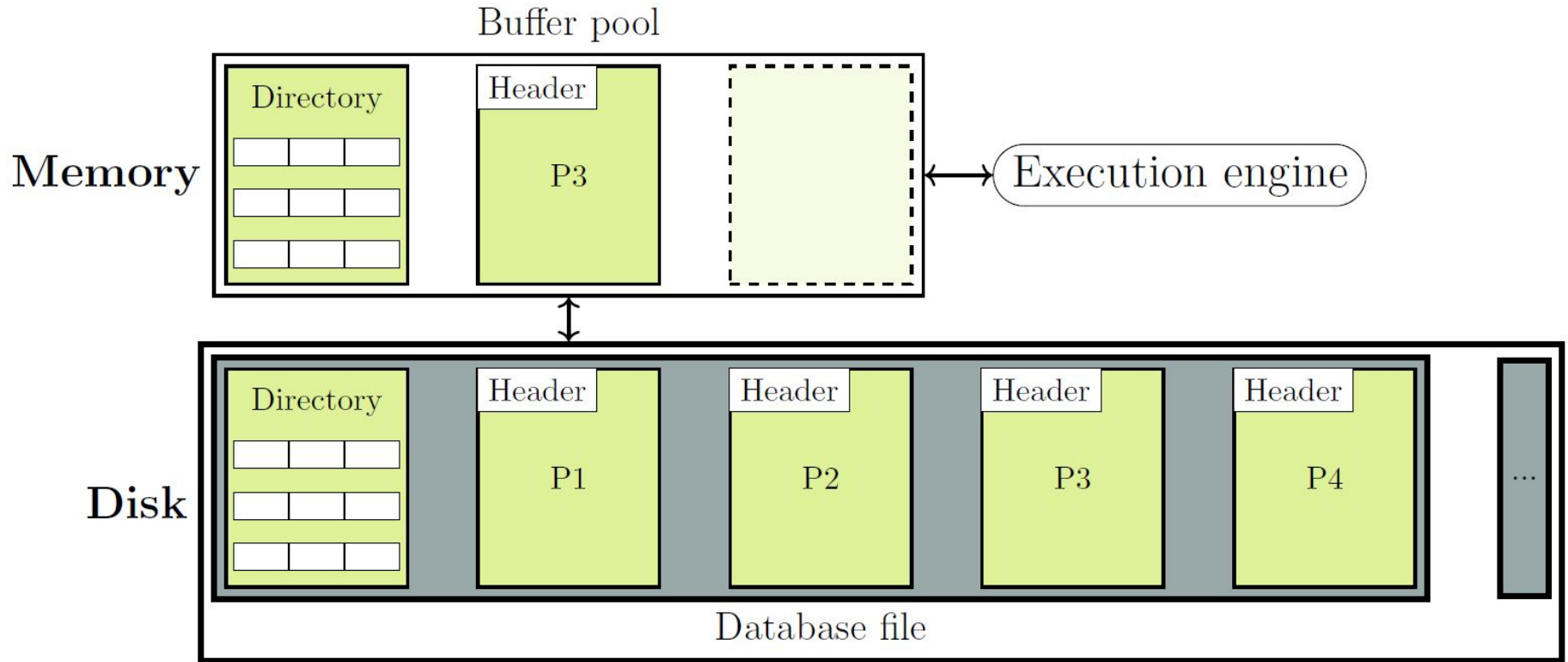
El diseño físico de una BD *disk-oriented* es...



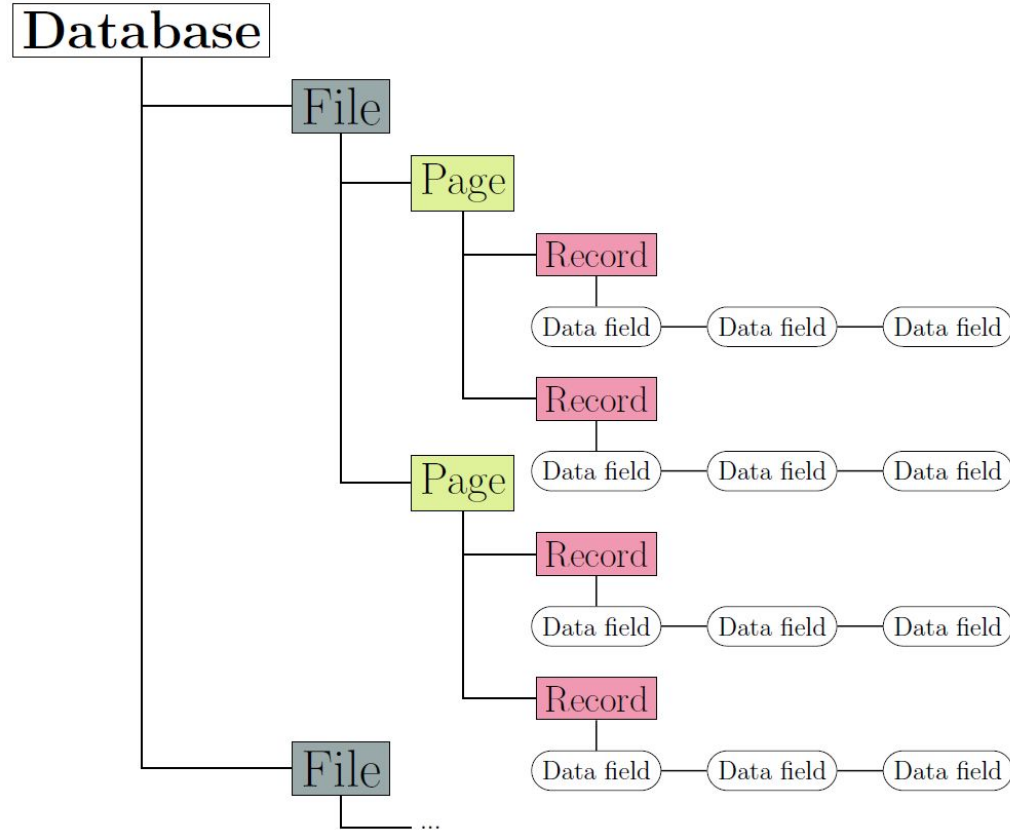
El diseño físico de una BD *disk-oriented* es...



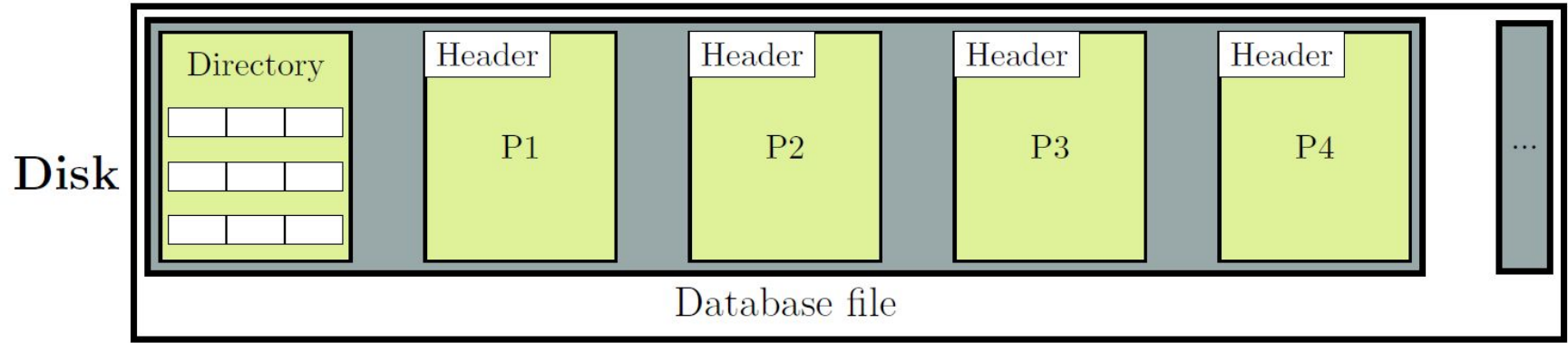
# El diseño físico de una BD *disk-oriented* es...



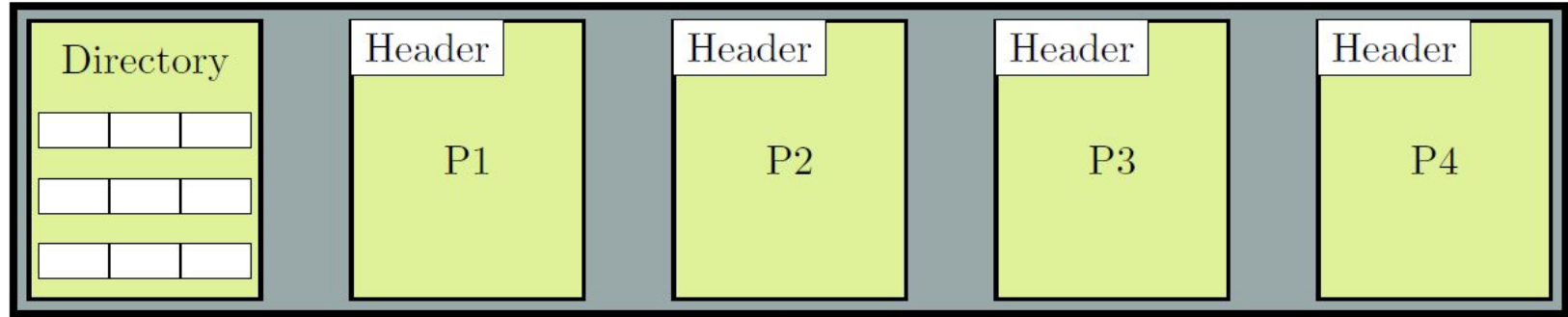
# Se tiene la siguiente jerarquía de datos



# Las bases de datos se guardan en *archivos*



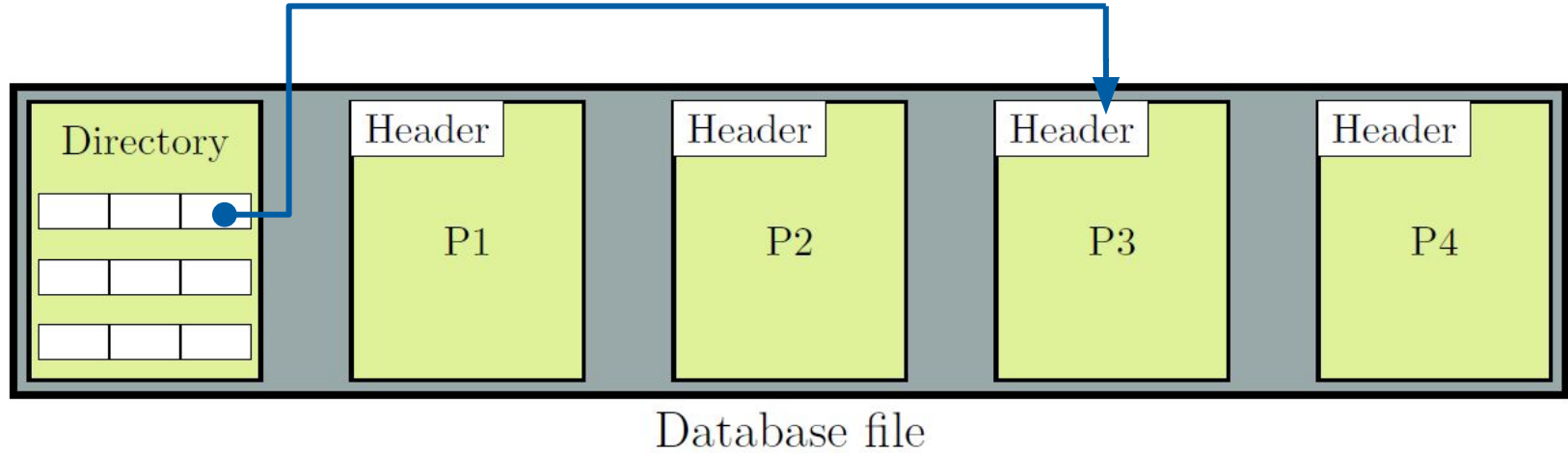
Cada archivo se divide en unidades lógicas de longitud fijas denominadas *páginas*



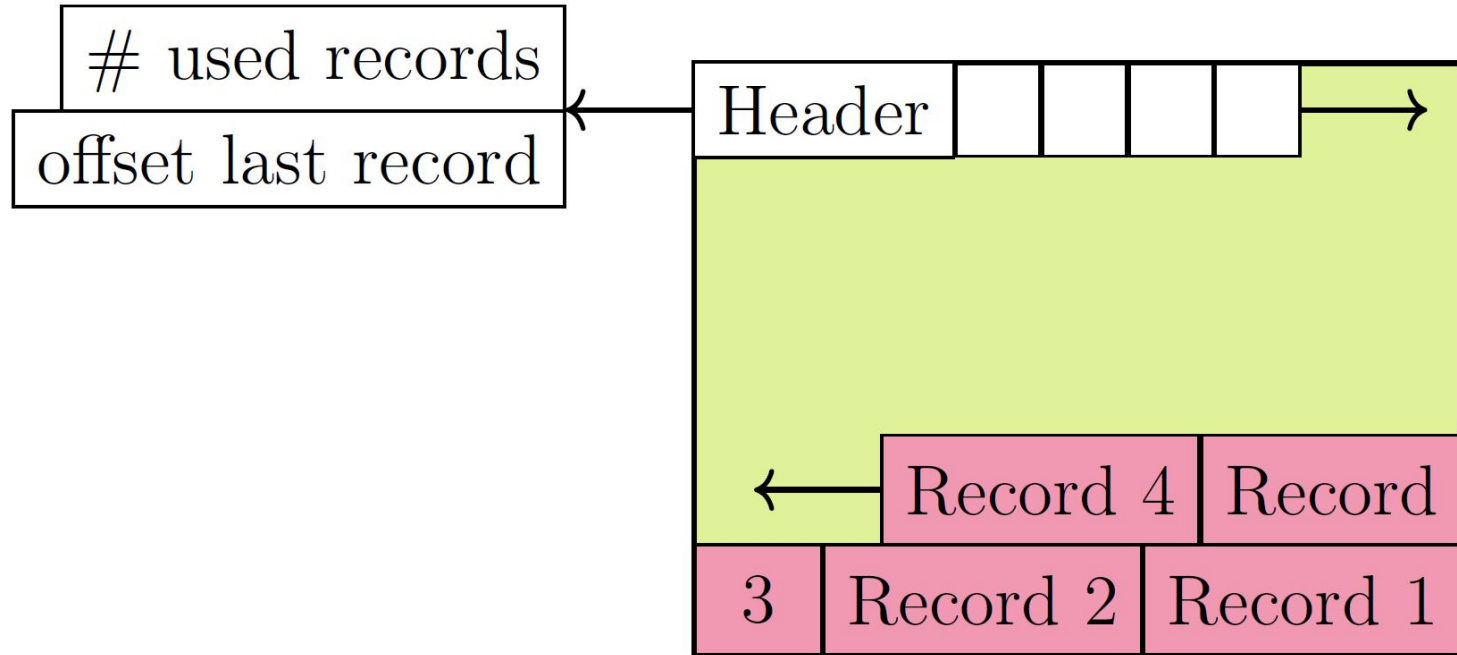
Database file



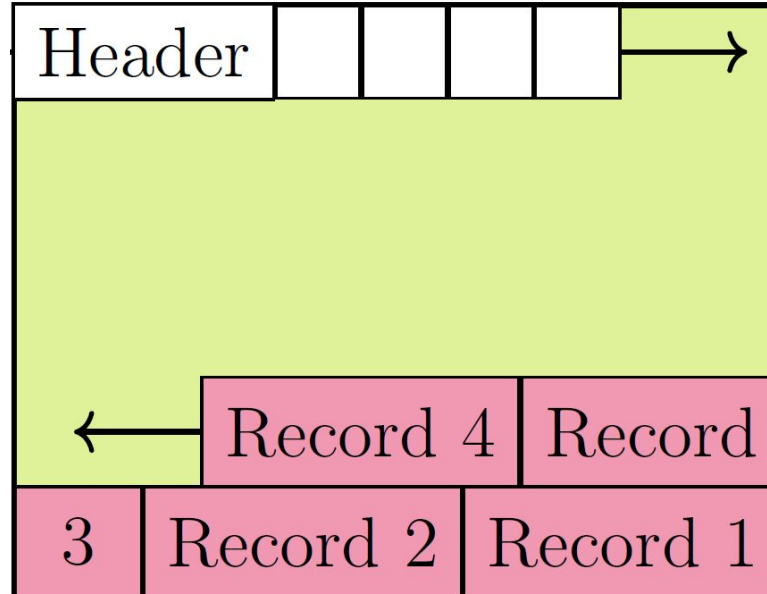
La manera más frecuente para encontrar páginas es con el directorio de páginas



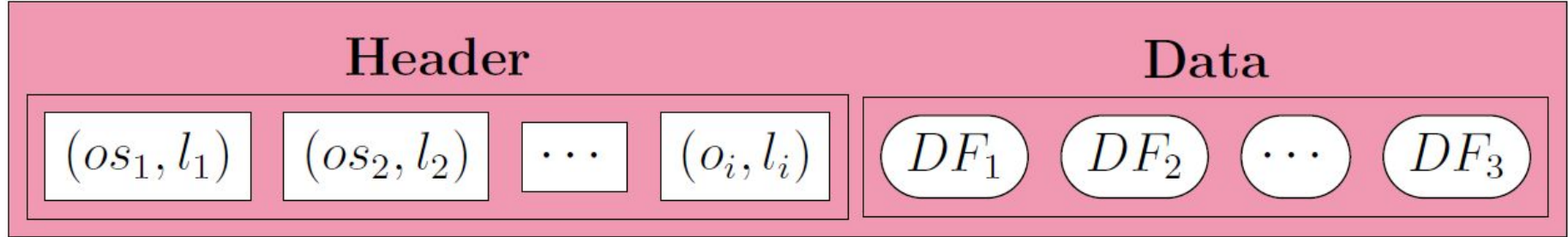
Los registros de las páginas se guardan generalmente utilizando *slotted-page structure*



Cada registro guarda una colección de ítems de datos de distintas columnas de una relación



Generalmente, se guardan los registros con longitud variable



Los registros pueden no ser guardados en varias páginas  
(*unspanned*)

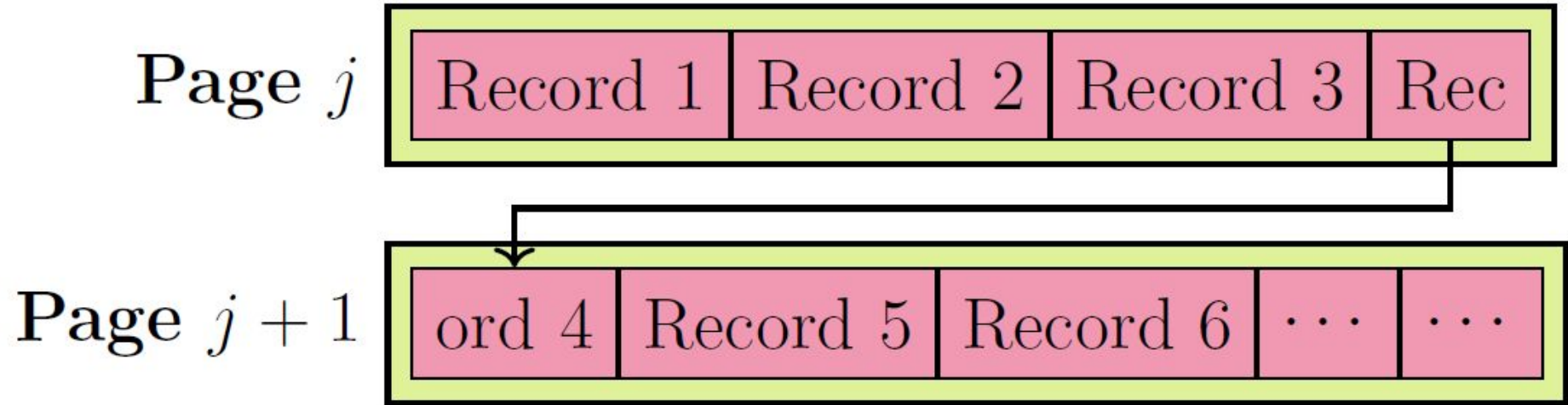
Page  $j$



Page  $j + 1$



... O se pueden guardar un registro en varias páginas  
(*spanned*)



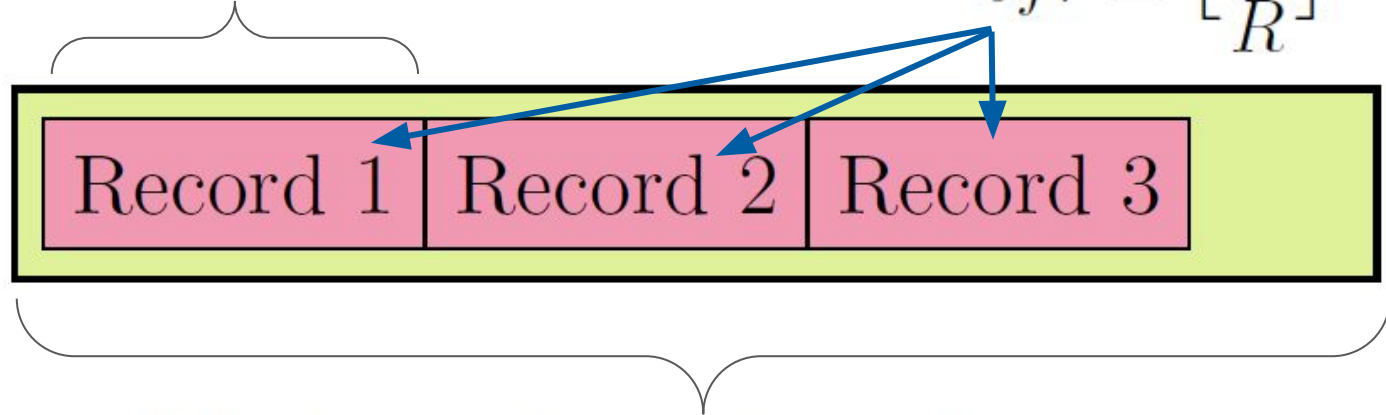


Se puede calcular para los páginas *unspanned* la cantidad de de registros que entran dentro de una página (*blocking factor*)

$R$  is the number of bytes for a record.

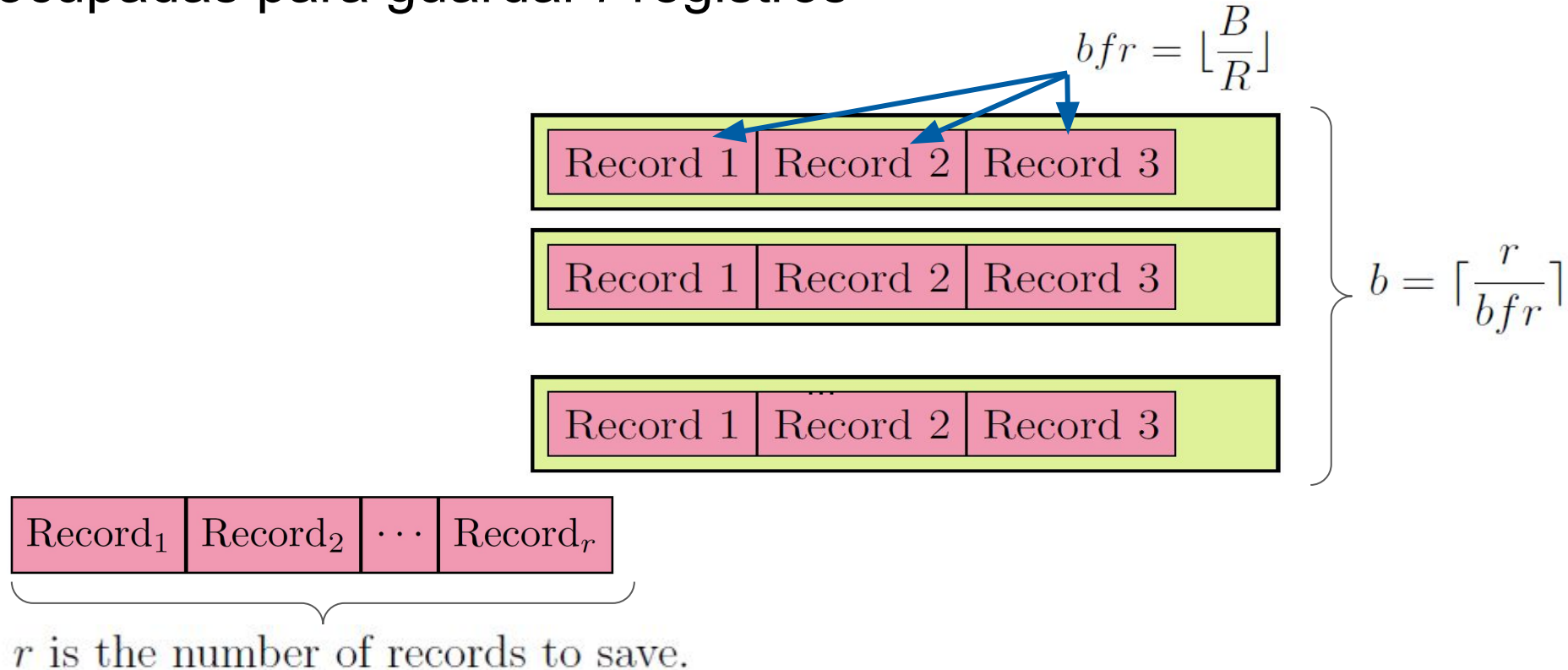
$$bfr = \lfloor \frac{B}{R} \rfloor$$

Page  $j$

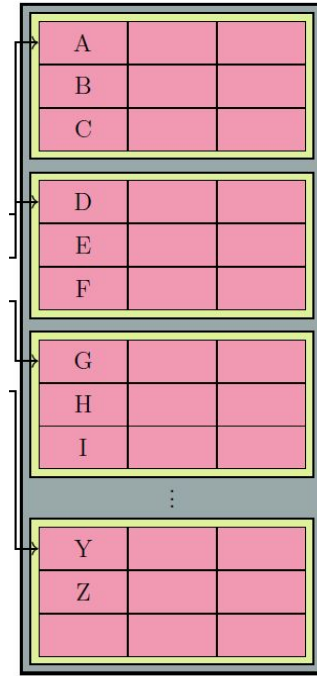


$B$  is the number of bytes for a page.

Se puede calcular la cantidad la cantidad de páginas  $b$  ocupadas para guardar  $r$  registros



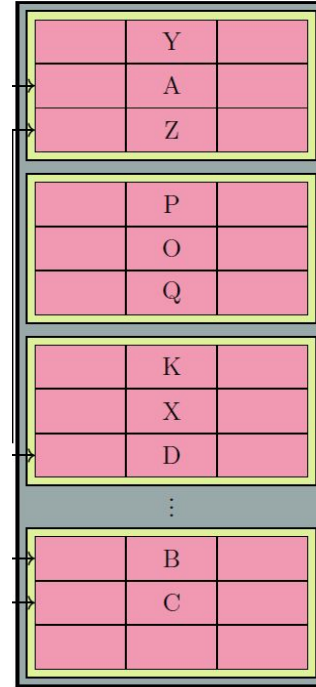
# La cantidad de páginas que se buscan depende el orden de los registros



File

**Ordenado**

**Busqueda binaria  $O(\log n)$**



File

**No ordenado**

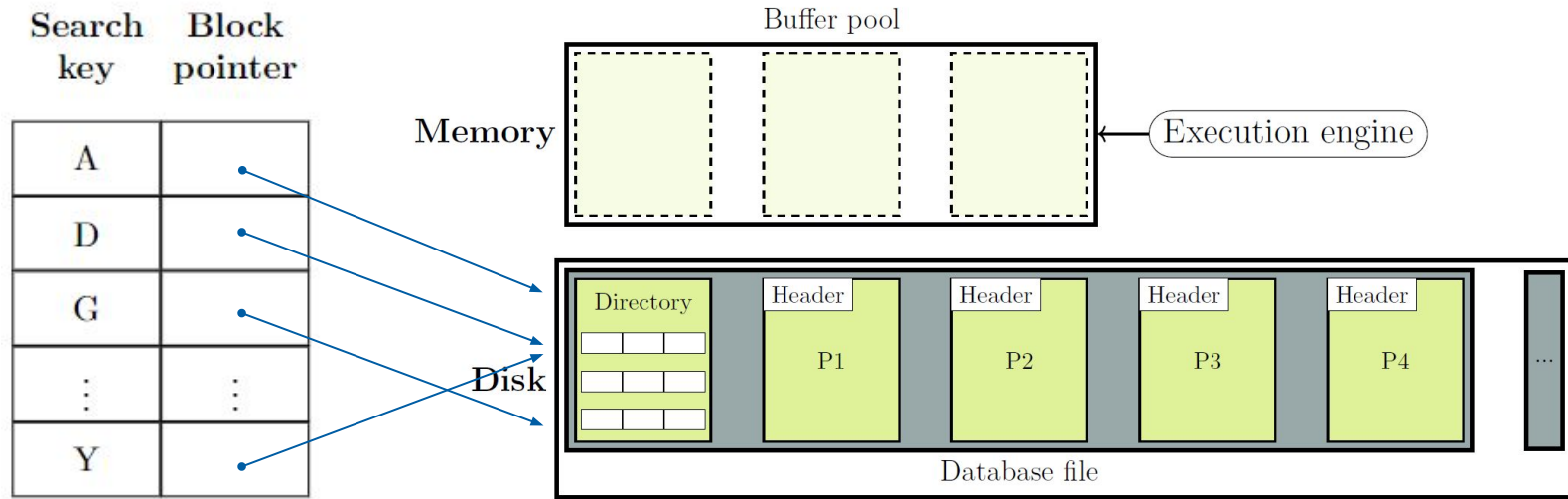
**Búsqueda lineal avg  $O(n/2)$**

# Se puede ver la información del almacenamiento físico en SQL Server

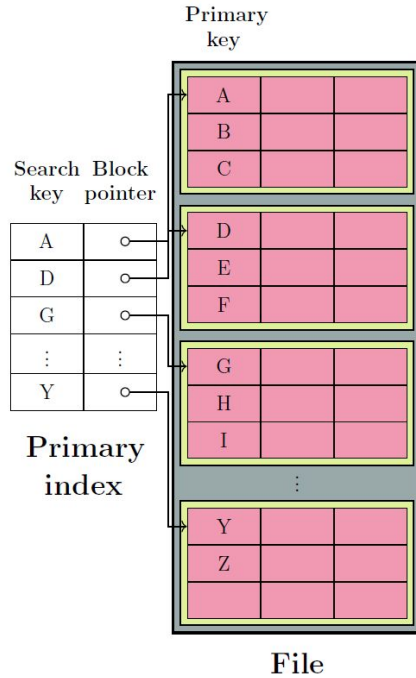


```
SELECT sys.fn_PhysLocFormatter(%%physloc%%) AS [File:Page:Slot], *  
FROM T;
```

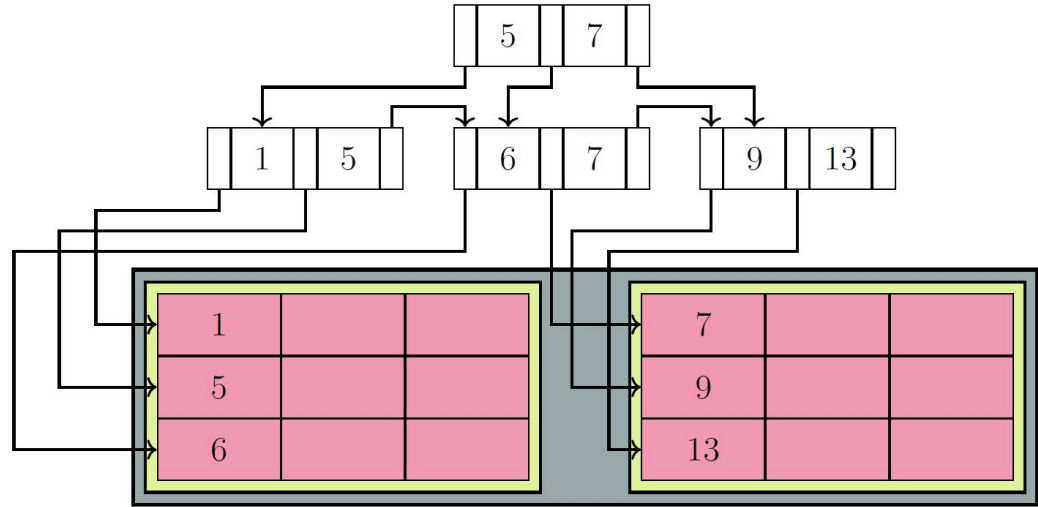
# Los DBMS utilizan estructuras auxiliares para realizar búsquedas de registros más rápidas



# Existen dos tipos de estructuras auxiliares

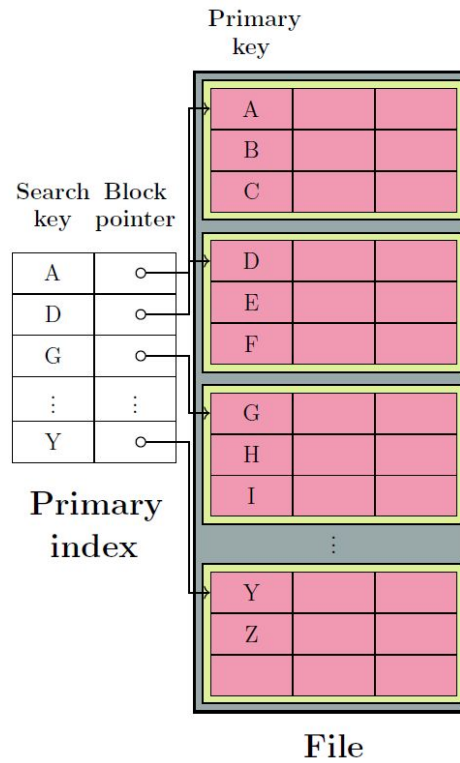


**Ordenados**



**Multinivel**





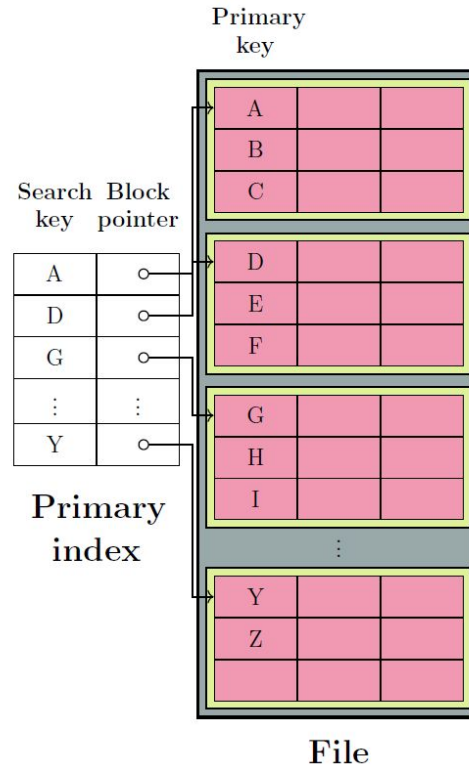
## Indices ordenados

Los índices son como bibliotecas, que utilizan una llave para buscar donde se encuentra un libro

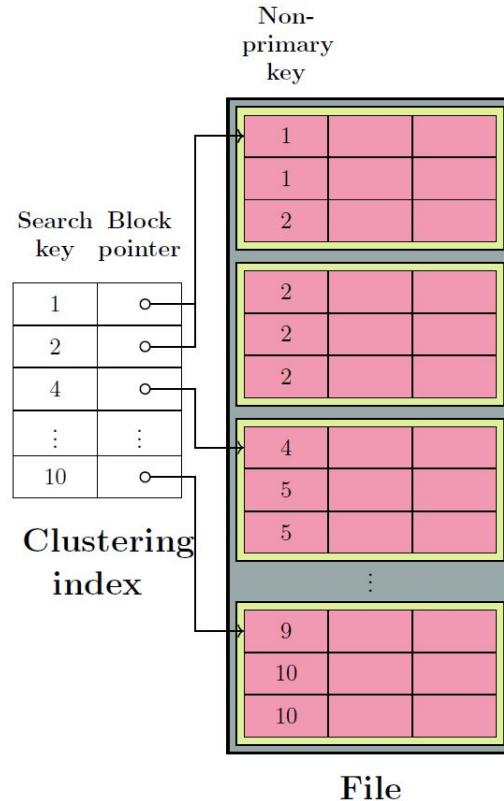


Existen diversos tipos de índices ordenados

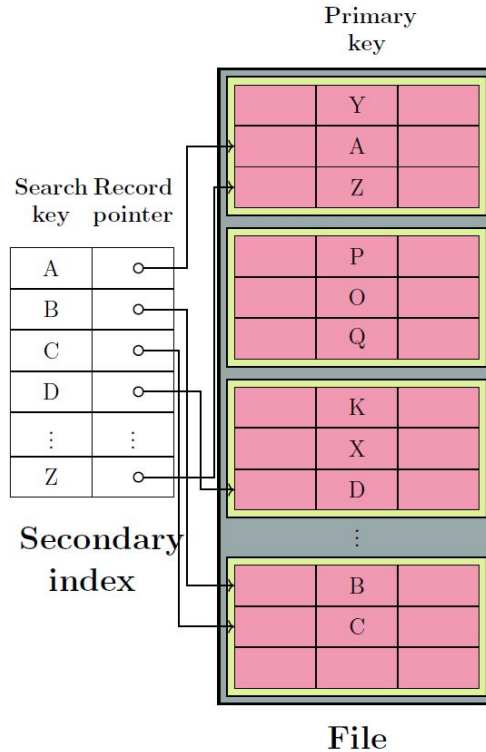
Están los índices primarios, que crean un índice basado en una llave primaria con los datos ordenados por ella.



Están los índices clustered, que crean un índice basado en atributos no llave, con los datos ordenados por el atributo.

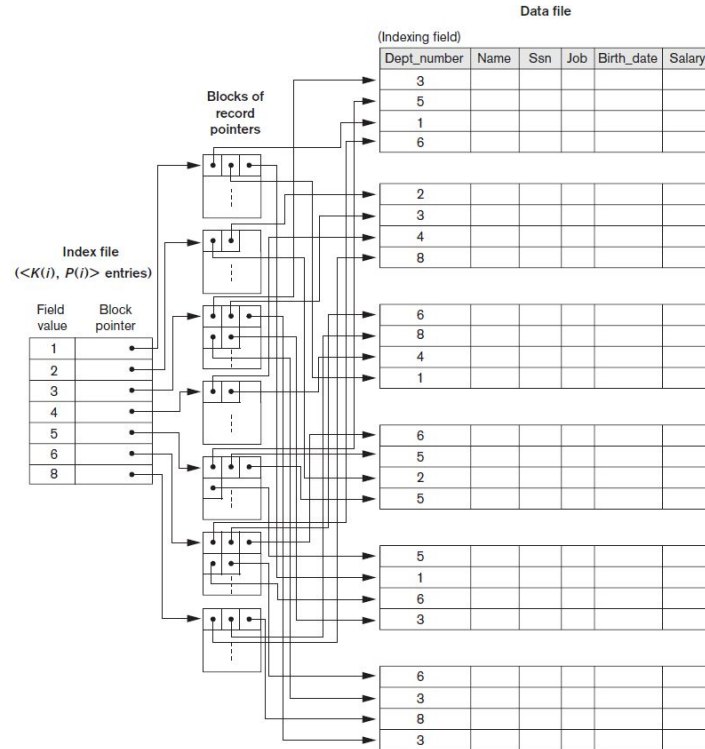


Están los índices secundarios (llave) con un índice para datos que no se encuentran ordenados por la llave.





Están los índices secundarios (llave) con un índice para datos que no se encuentran ordenados por la llave.



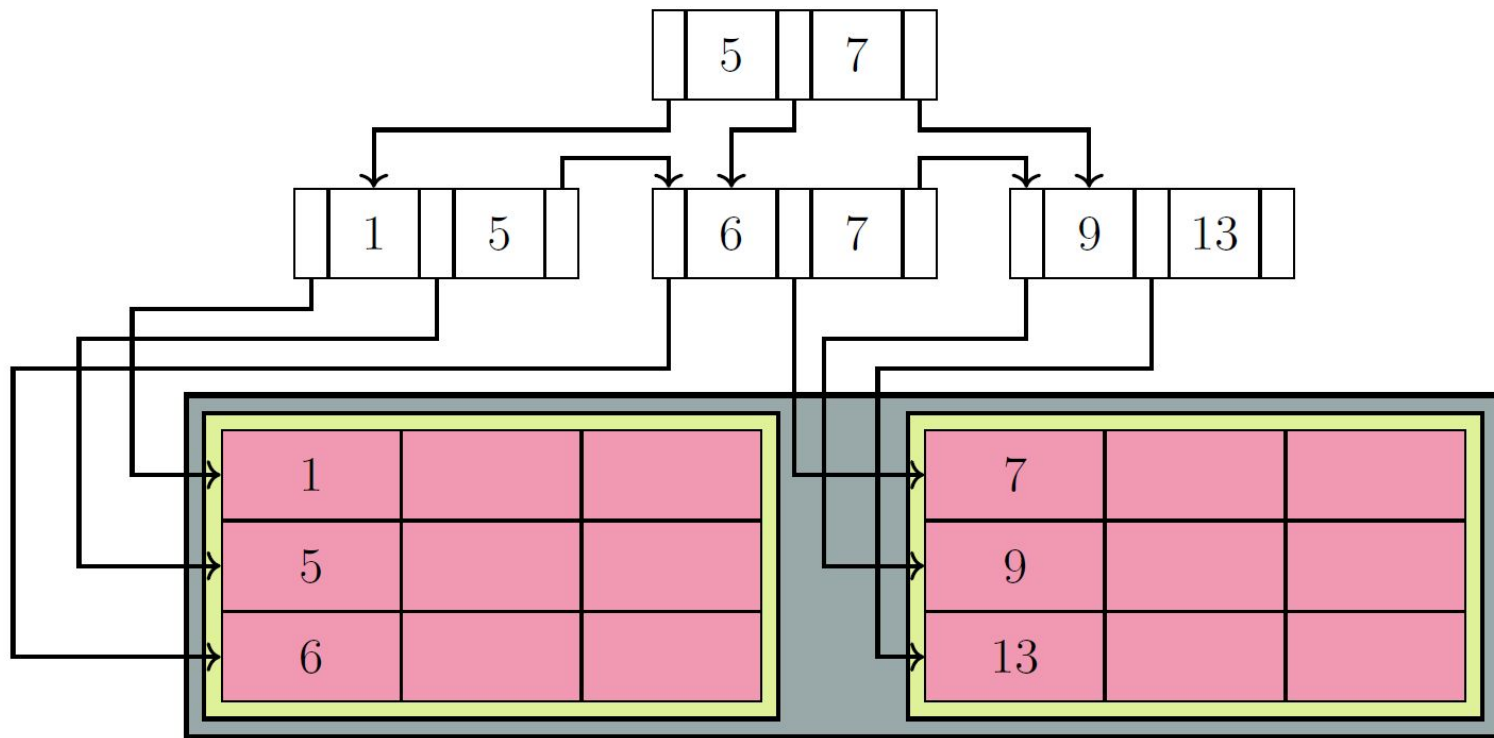
¿Cuántos índices primarios se pueden tener?

¿Cuántos índices clustering se pueden tener?

¿Se puede tener un índice primario y clustering a la vez?

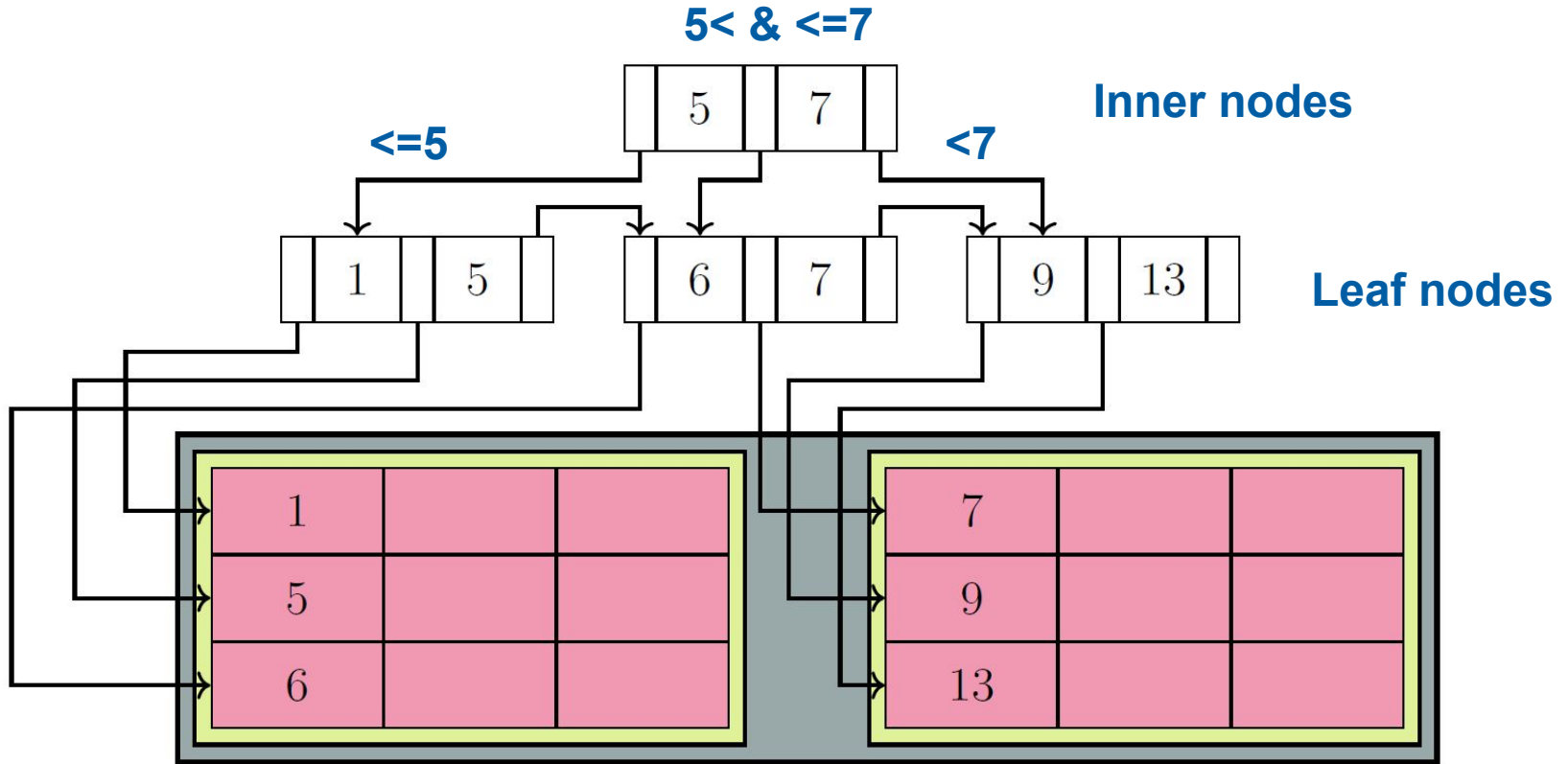
¿Cuántos índices secundarios se pueden tener?

¿Se puede tener un índice secundario cuando hay uno primario o de clustering?



## Indices multinivel

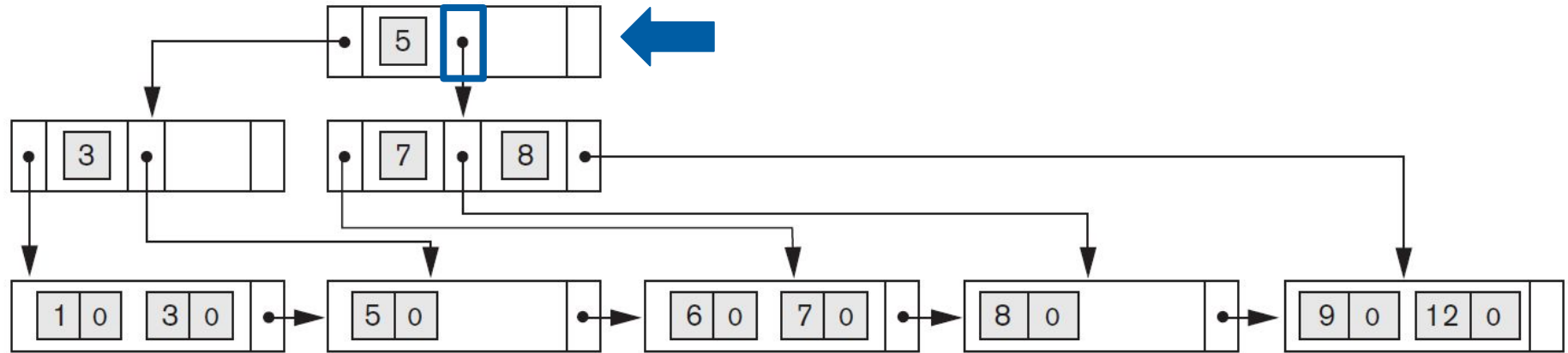
# Los DBMS principalmente usan B+ trees



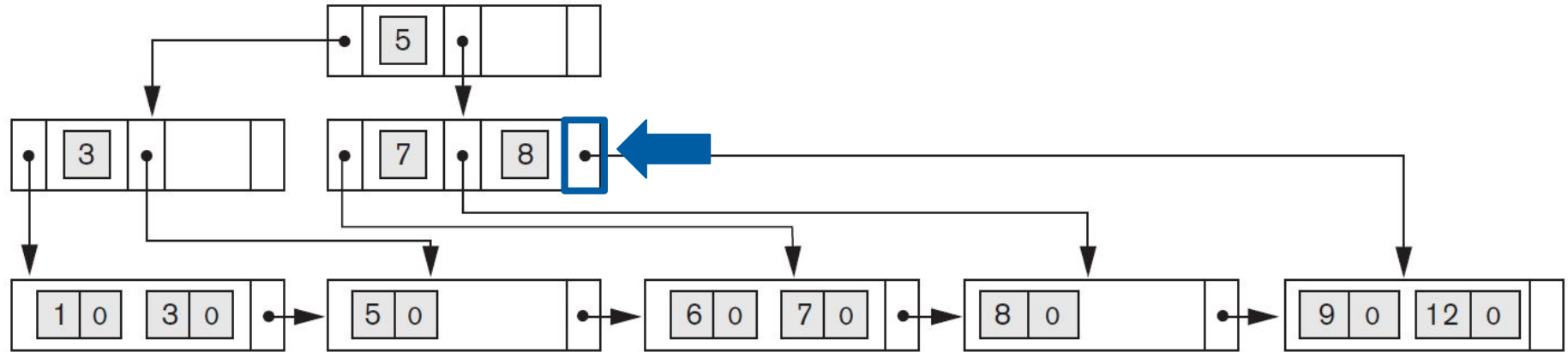


# Buscar en B+ Trees

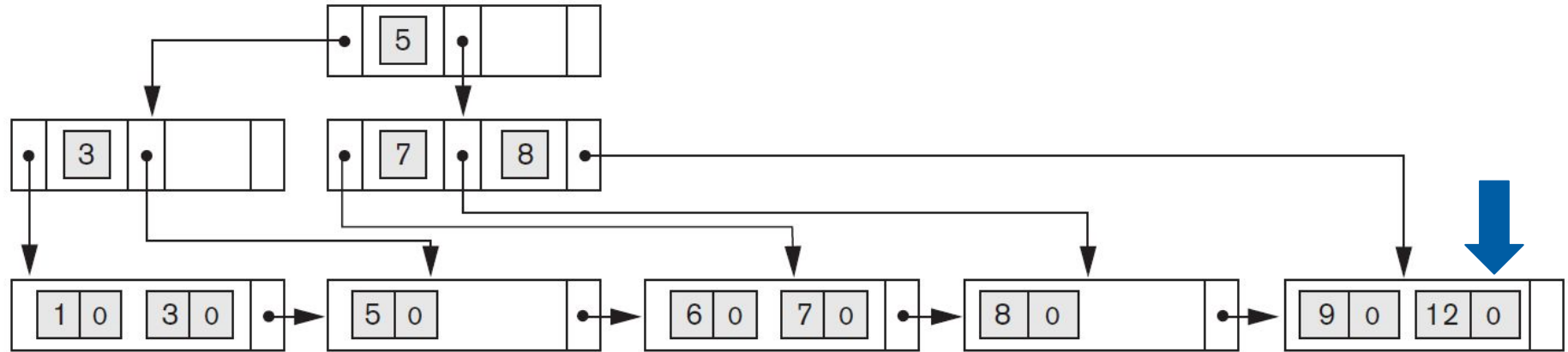
Buscar 9



## Buscar 9



## Buscar 9



**Algorithm 1** Searching for a record that has a search key value  $v$  as the index of a B+ Tree of order  $p$

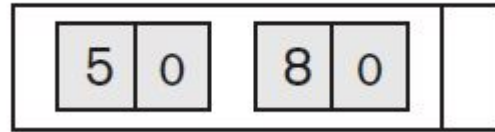
---

```
function FIND( $v$ )
   $n = \text{tree.getRootPage}();$ 
   $n.\text{readPage}();$ 
   $p = n.\text{getNumberPointers}();$ 
  while ( $\neg n.\text{isLeafNode}()$ ) do
     $l = n.\text{getLarger}(v);$   $\triangleright$  The node  $n$  with  $i$   $x_i$  values, get those that are  $v \leq x_i$ 
     $xi, pi = l.\text{min}();$   $\triangleright$  Finds the smallest  $x_i$  that is larger than  $v$  with the respective pointer  $pi$ 
    if ( $xi.\text{isNull}()$ ) then  $\triangleright v > x_i$ 
       $n = n.\text{getLastPointer}();$   $\triangleright$  Gets last non-null pointer of  $n$ 
    else if ( $v == xi$ ) then  $\triangleright v = x_i$ 
       $n = n.\text{getNext}(pi);$   $\triangleright$  Gets the pointer following  $pi$ 
    else  $\triangleright v < x_i$ 
       $n = pi;$   $\triangleright$  Gets the current pointer
    end if
     $n.\text{readPage}();$ 
  end while
   $r = n.\text{hasRecordWithKey}(v)$ 
  if ( $\neg r.\text{isNull}()$ ) then  $\triangleright$  We did found a record with the value
    return  $n;$ 
  else  $\triangleright$  We did not found a record with the value
    return  $null;$ 
  end if
end function
```

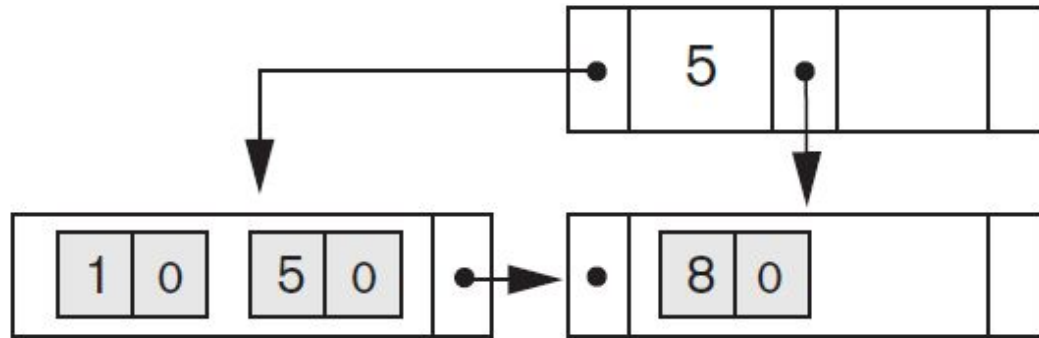
---

# Insertar en B+ Trees

Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6

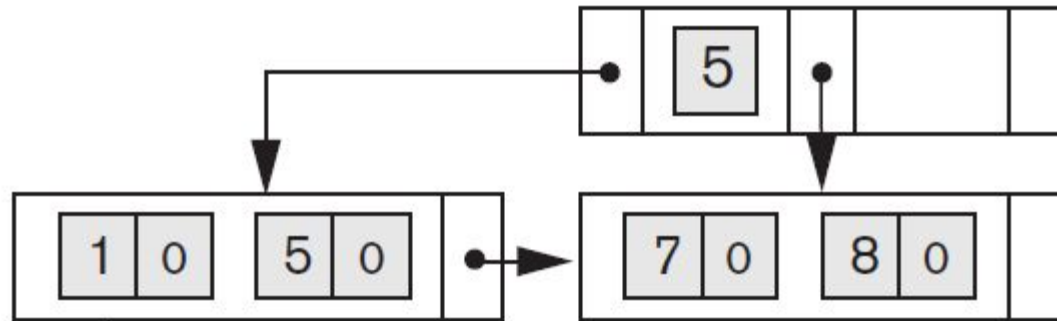


Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6

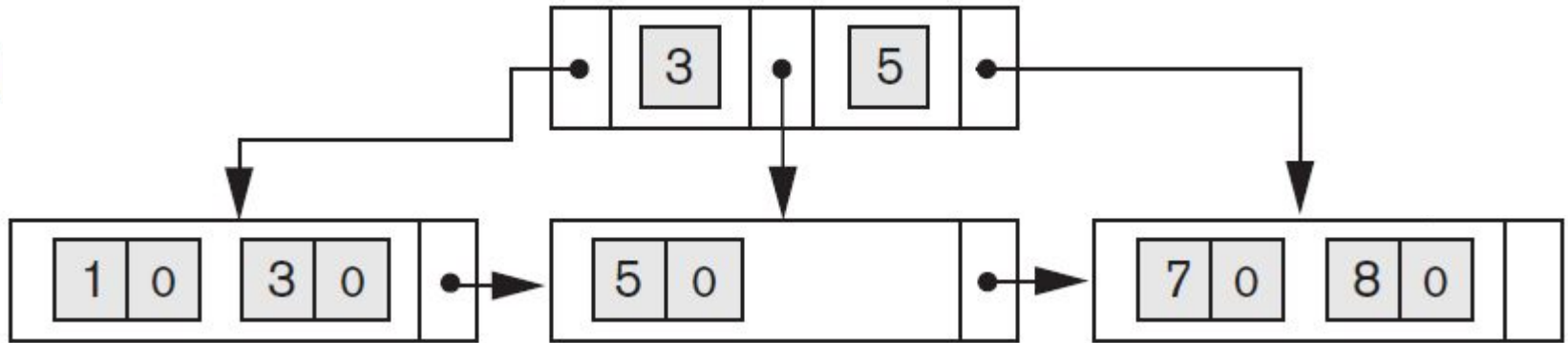




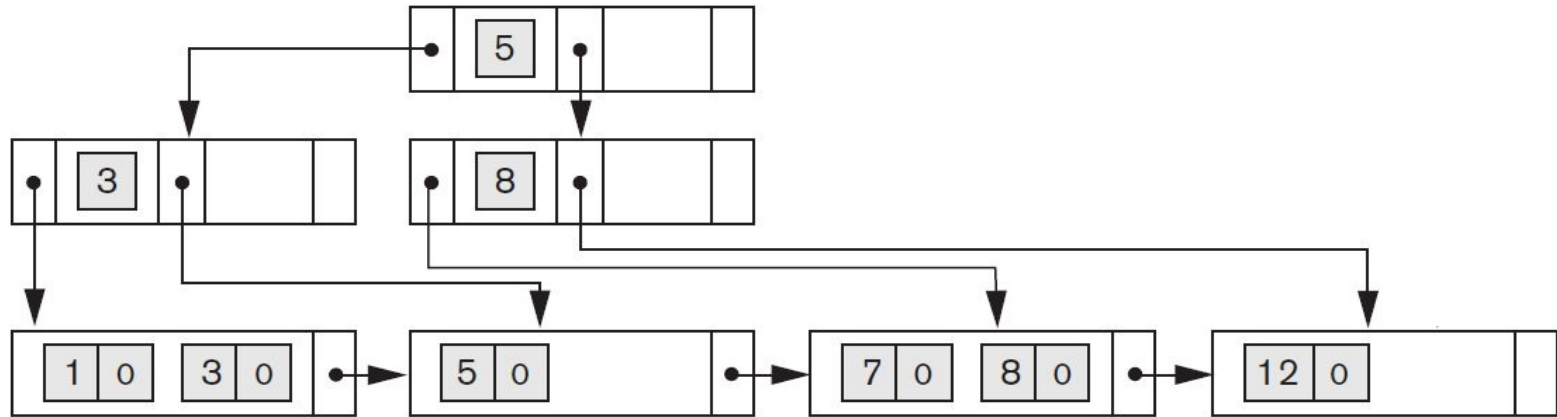
Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6



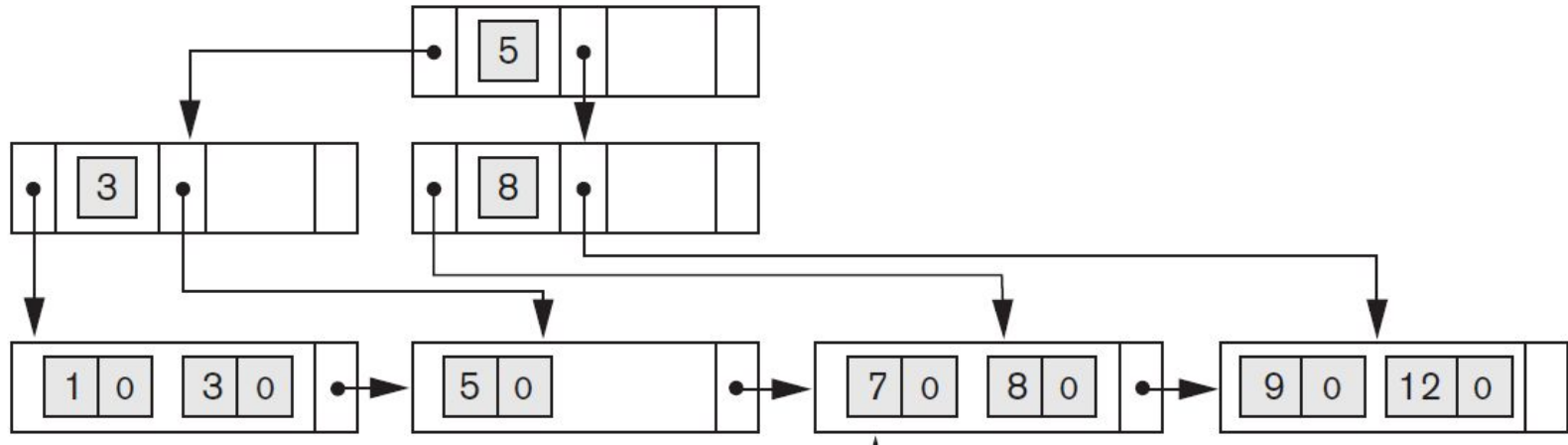
Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6



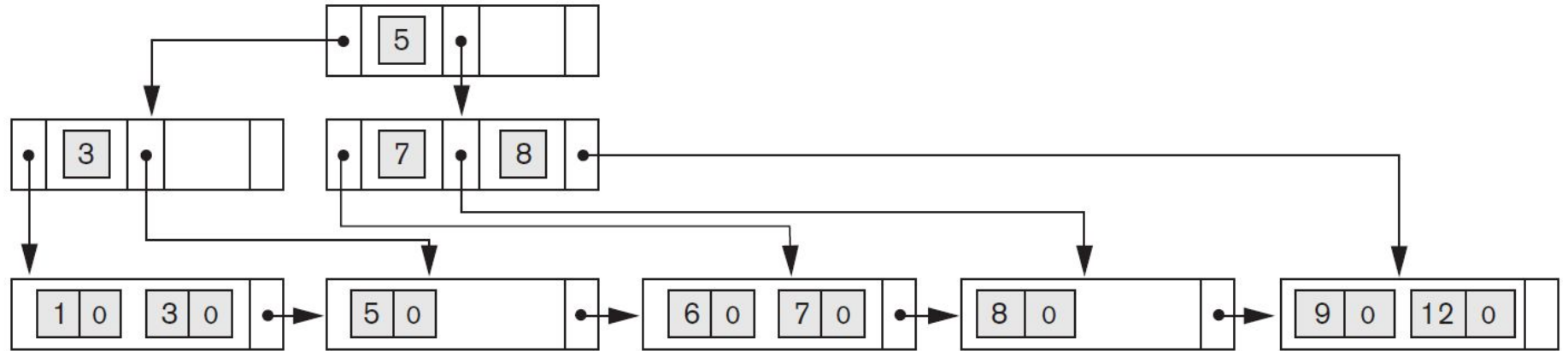
Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6



Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6

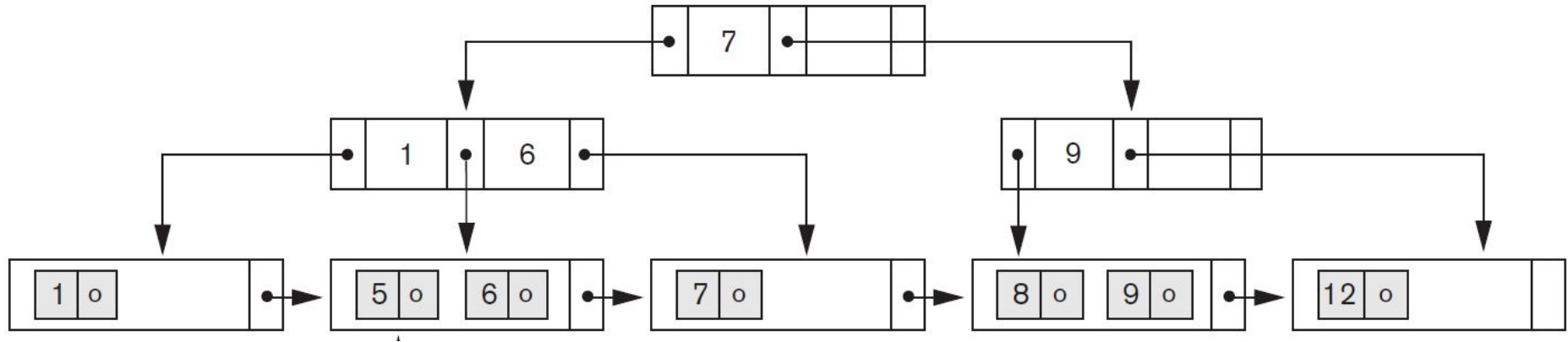


Insertion sequence: 8, 5, 1, 7, 3, 12, 9, 6

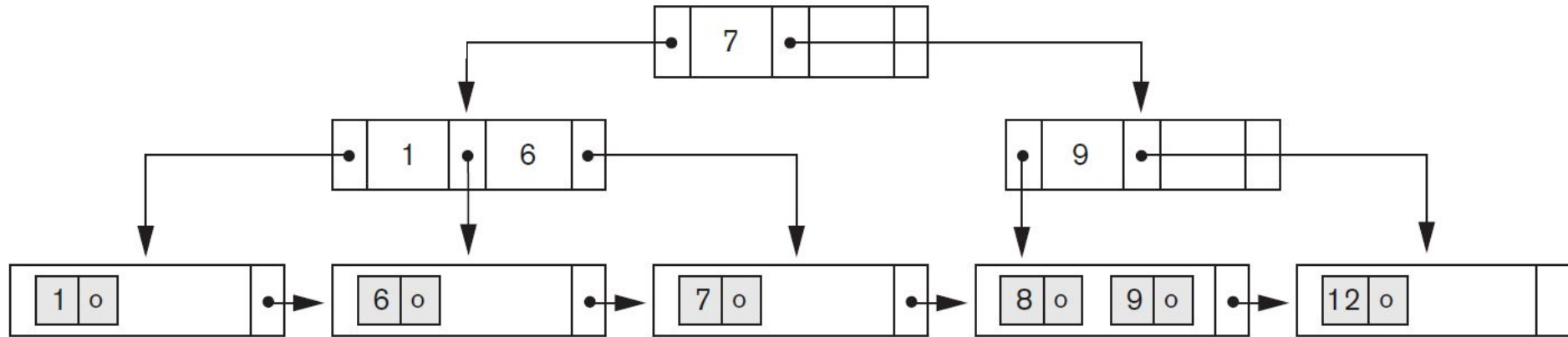


# Borrar en B+ Trees

Deletion sequence: 5, 12, 9

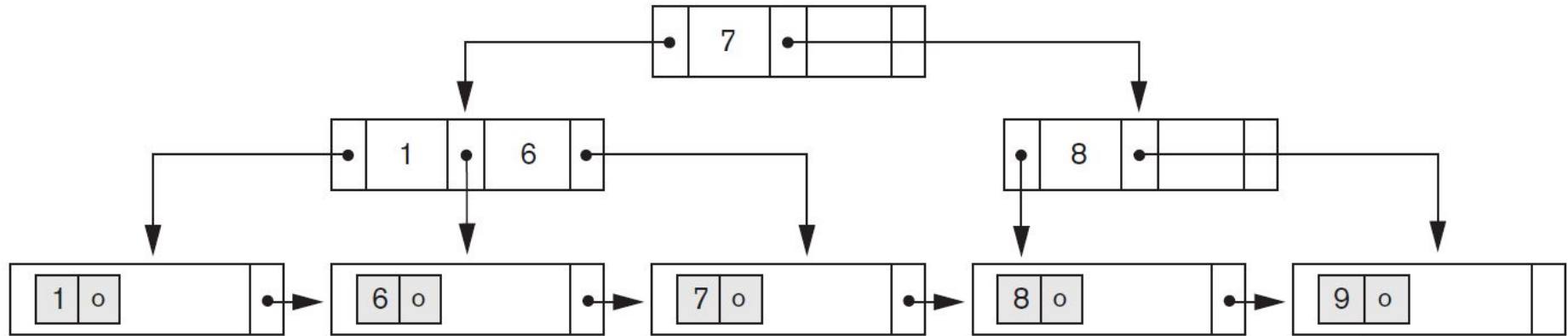


Deletion sequence: 5, 12, 9

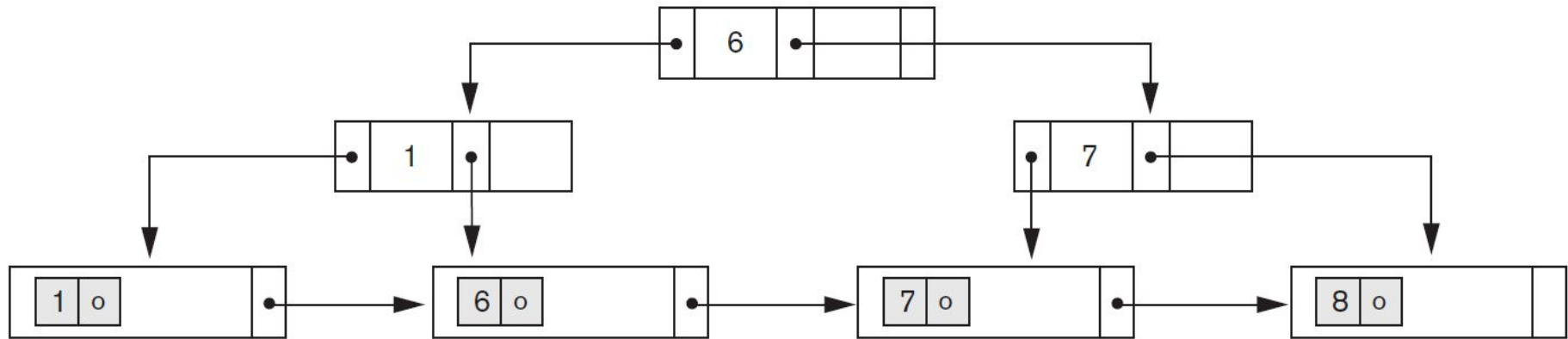




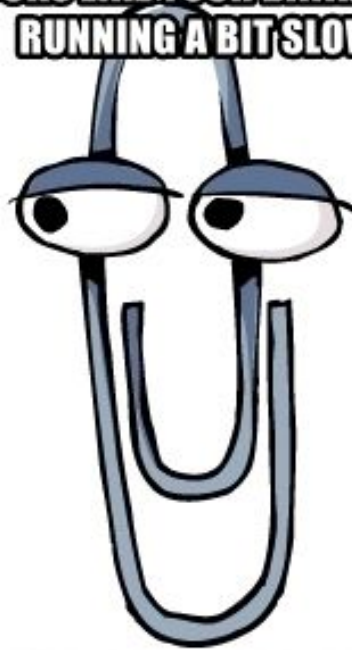
Deletion sequence: 5, 12, 9



Deletion sequence: 5, 12, 9



**IT LOOKS LIKE YOUR DATABASE IS  
RUNNING A BIT SLOW**



**WOULD YOU LIKE TO RUN THE DATABASE  
TUNING ADVISOR?**

generator.net

# Referencias

- R. Elmasri and S. Navathe, Fundamentals of database systems, 7th ed. Pearson, 2016, chapters 16 and 17.
- A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY: McGraw-Hill, 2020, chapter 12, 13 and 14.
- A. Crotty and L. Ma. Lecture #3, #4, #5,#6 and #7. [Online]. Available: <https://15445.courses.cs.cmu.edu/fall2021/schedule.html>
- Microsoft. Sql server guides. [Online]. Available: <https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-guides>
- Microsoft. Database files and filegroups. [Online]. Available: <https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-files-and-filegroups>