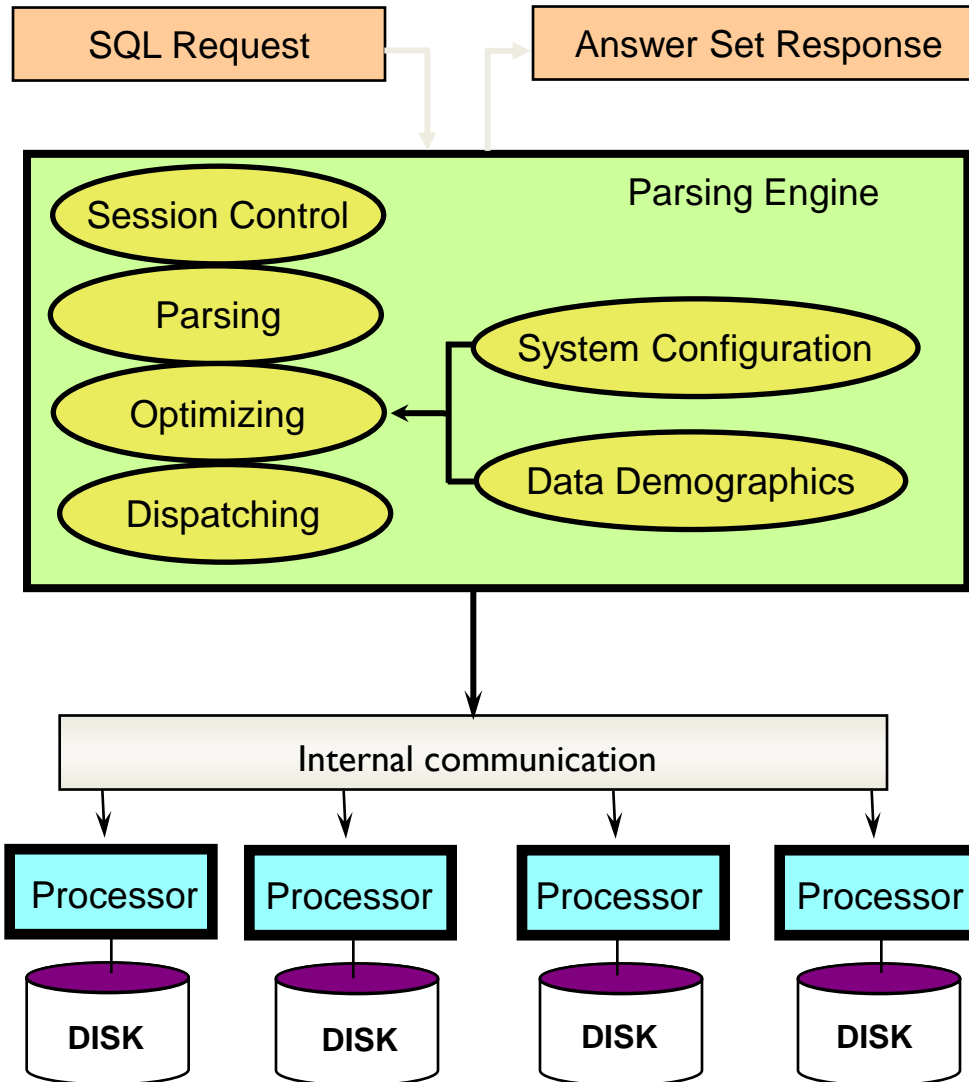


# Optimización



# Organización de Archivos

Una BD se almacena como una colección de archivos

Cada archivo contiene **registros del mismo tipo** y se divide en **bloques de igual tamaño**

**Organización de Archivos** se refiere a la forma en que los datos son almacenados dentro de un archivo y las formas en que pueden accederse

Dos tipos de archivos clásicos:

**HeapFile**

**SortedFile**

# HeapFile

## Registros sin orden

Al insertarse un registro, se lo agrega al final del archivo o en alguno de los bloques con espacio libre

Las operaciones de búsqueda requieren búsqueda lineal por **todos los bloques** del archivo

# SortedFile

**Registros ordenados** a partir de una clave de búsqueda **A**

Al insertarse un registro se lo agrega ordenadamente, lo que puede provocar una reorganización en los bloques del archivo

**Mejoran las búsquedas por A**, pero el resto de las operaciones suelen requerir una búsqueda lineal

Estructuras adicionales: **aceleran** ciertas operaciones de búsqueda sobre tablas

Mayor costo en operaciones de escritura, actualización y borrado

Mayor costo en espacio ocupado

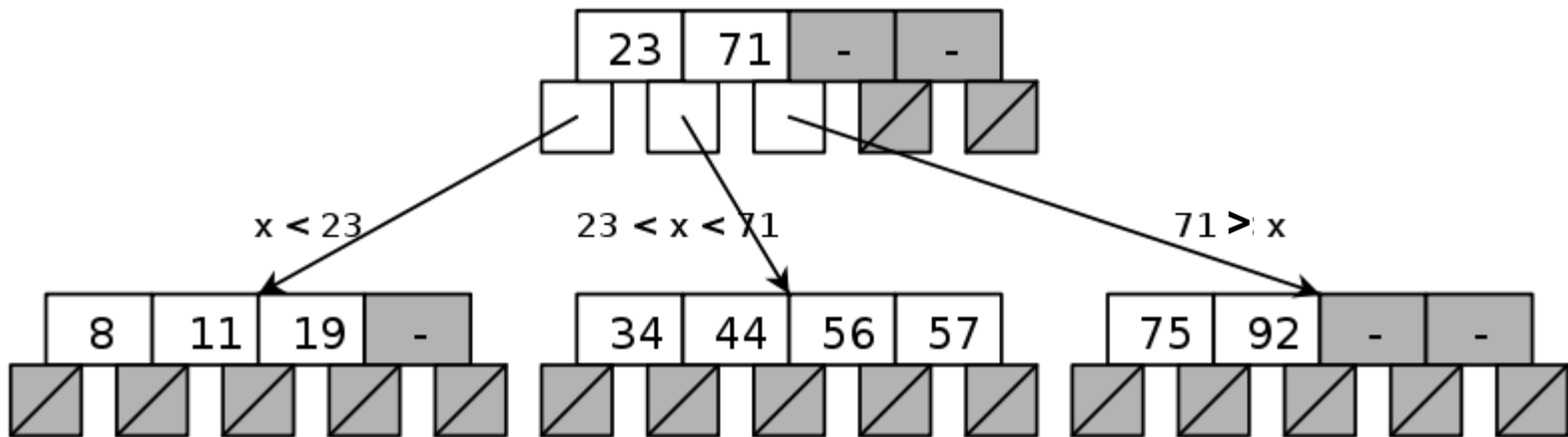
Dos tipos de índices clásicos son:

**B+**

*Hash*

# Índices B+

Los índices B+ son árboles balanceados



## SUMARIO

PRÓLOGO  
10

INTRODUCCIÓN  
11

CAPÍTULO UNO  
*La astronomía a través de los tiempos*  
12

CAPÍTULO DOS  
*Estrellas y galaxias*  
24

CAPÍTULO TRES  
*Instrumentos y técnicas*  
52

CAPÍTULO CUATRO  
*Entender los cambios del cielo*  
78

CAPÍTULO CINCO  
*Una Guía del cielo*  
100

CAPÍTULO SEIS  
*Un viaje por el Sistema Solar*  
228

CAPÍTULO SIETE  
*Sondear el universo*  
264

DIRECTORIO DE RECURSOS  
273

## Directorio de recursos

# ÍNDICE y GLOSARIO

**E**n esta combinación de índice y glosario, los números de página en **negrita** indican la referencia principal y las *cursivas* señalan las ilustraciones y las fotografías.

## A

AAVSO 74

Abell, George 51

Achernar 103, 106, 121, 127, 129

Acrux 106, 163

Adams, John Couch 255, 255

adaptación a la oscuridad Proceso  
por el cual el ojo humano aumenta  
la sensibilidad en condiciones de  
poca (o ninguna) iluminación. 58

adaptadores de cámara 67

Adhara 106

Águila, Nebulosa del 212, 212

agujero negro Objeto macizo tan  
denso que no permite salir luz ni  
radiación alguna. 35, 35, 47, 267

Alpha ( $\alpha$ ) Tauri véase Aldebaran

Alpha ( $\alpha$ ) Ursa Minoris véase  
Polaris

Alpha ( $\alpha$ ) Virginis véase Spica

Altair 106, 112, 114, 115, 124, 126,  
137, 137, 164

altitud 63, 63, 81, 81

Ames, Adelaide 51

Andrómeda 29, 51, 108, 116,  
132-3, 132, 133

Andrómeda, Galaxia 28, 46, 48,  
49, 51, 133, 133

Anillo, Nebulosa del 43, 45, 186,  
186

Anillo, Cola de, Galaxia 162

anillos véase planetas

Antares 30, 91, 106, 183, 208, 209

golpean la atmósfera de la Tierra y  
hacen brillar algunos de sus gases.  
54, 98, 98

azimut 63, 63, 81, 81

## B

Baade, Walter 23

Barnard, E. E. 193

Barnard 33 véase Cabeza de Caballo,  
Nebulosa

Barnard, Estrella de 27, 30, 91,  
193

Bartsch, Jakob 189

Bayer, Johann 88-89, 153, 156,  
167, 173, 178, 179, 190, 196, 218,  
219, 226

Bellatrix 106

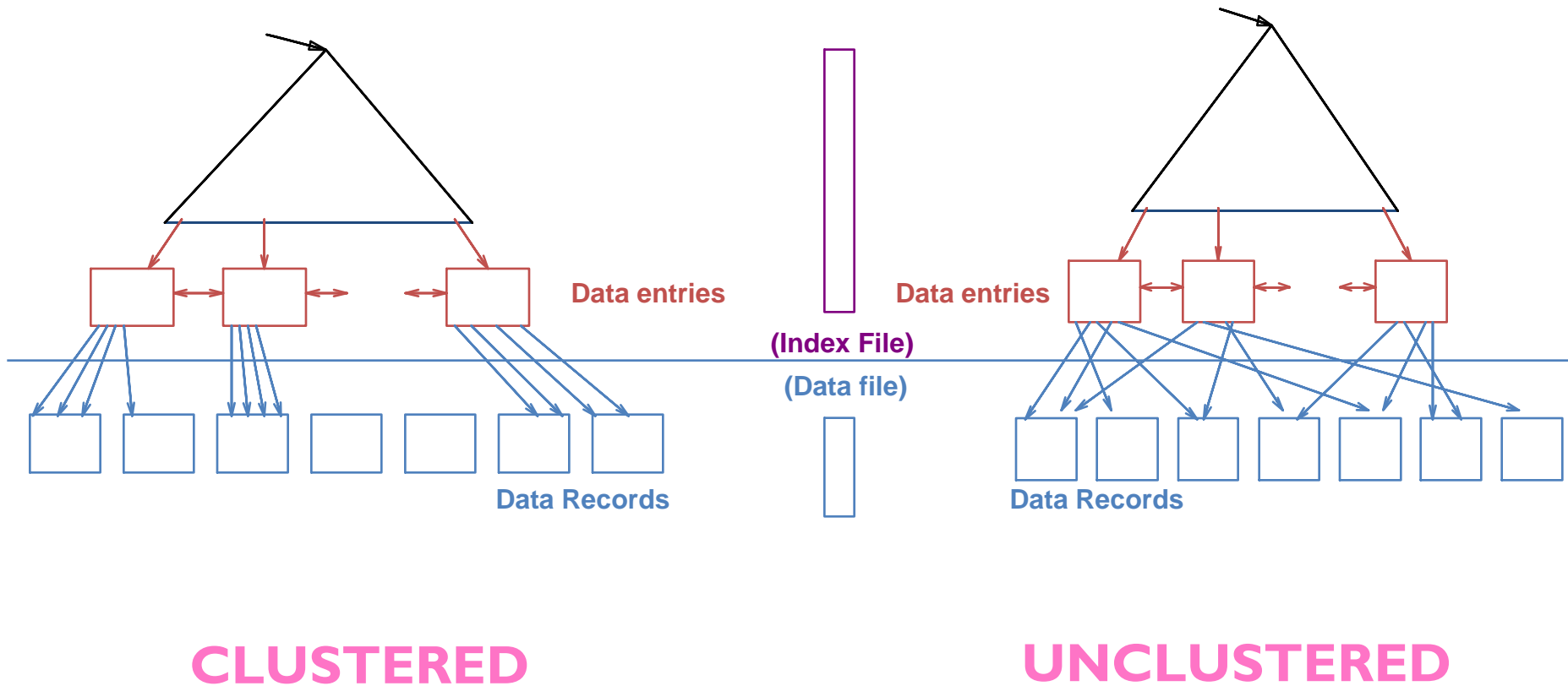
Berenice, Cabellera de véase Coma  
Berenices

Bessel, Friedrich 147

Beta ( $\beta$ ) Arae 138



# Clustered vs. Unclustered Index



# Índices Hash

**CORREO ARGENTINO** 10 AÑOS ABRAZANDO LA PATRIA  
CORREO OFICIAL

INICIO MAPA DEL SITIO CONTACTO PREGUNTAS FRECUENTES  
ACCESO CLIENTES

CORREO ARGENTINO PRODUCTOS Y SERVICIOS FILATELIA CORREO Y LA COMUNIDAD NOTICIAS

INICIO » CONSULTA DE CPA

## Consulta de CPA

Ingresá a continuación la dirección sin acentos ni apóstrofes para conocer su código postal.

**Provincia** Ciudad Autonoma de Buenos Aires **Localidad** Ciudad Autonoma Buenos Aires

**Calle** Tinogasta **Altura** 2950

KK2ADA CAPTCHA Consultar

**CALLE TINOGASTA 2950 C1417EHL**

- SEGUIMIENTO DE ENVÍOS
- CONSULTA CPA
- ATENCIÓN AL CLIENTE
- SUCURSALES

**CORREO ARGENTINO** 10 AÑOS ABRAZANDO LA PATRIA  
CORREO OFICIAL

INICIO MAPA DEL SITIO CONTACTO PREGUNTAS FRECUENTES  
ACCESO CLIENTES

CORREO ARGENTINO PRODUCTOS Y SERVICIOS FILATELIA CORREO Y LA COMUNIDAD NOTICIAS

INICIO » CONSULTA DE CPA

## Consulta de CPA

Ingresá a continuación la dirección sin acentos ni apóstrofes para conocer su código postal.

**Provincia** Ciudad Autonoma de Buenos Aires **Localidad** Ciudad Autonoma Buenos Aires

**Calle** Tinogasta **Altura** 3000

BDGS4Y CAPTCHA Consultar

**CALLE TINOGASTA 3000 C1417EHL**

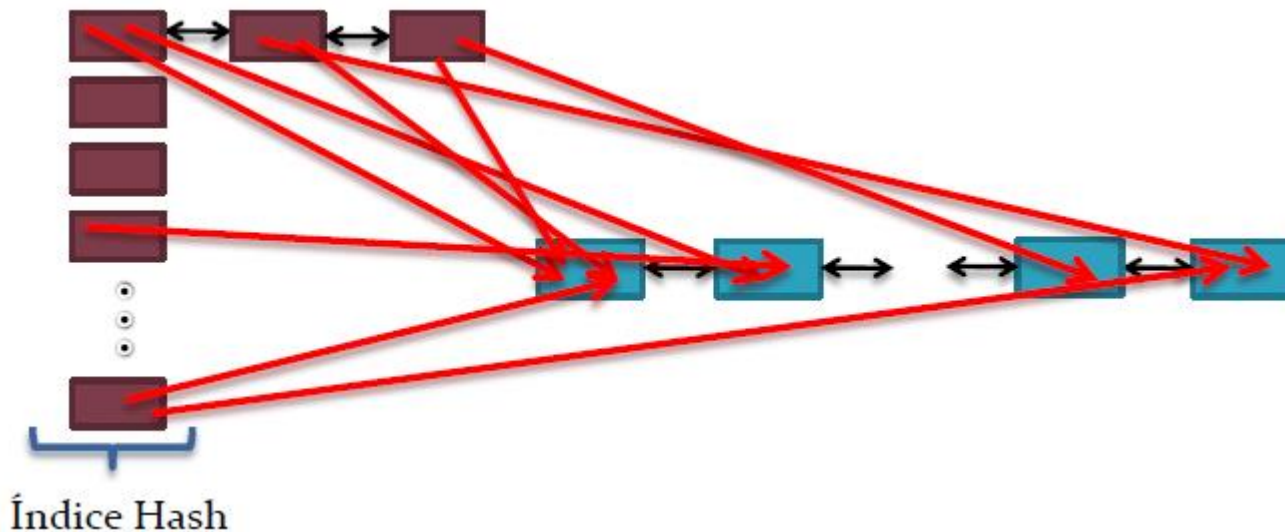
- SEGUIMIENTO DE ENVÍOS
- CONSULTA CPA
- ATENCIÓN AL CLIENTE
- SUCURSALES

# Índices Hash

**Una tabla de hash almacena las claves de búsqueda**

Cada posición de una tabla hash se asocia con un conjunto de registros. Por esta razón cada posición suele llamarse “un bucket”, y los valores de hash, “ índices bucket”.

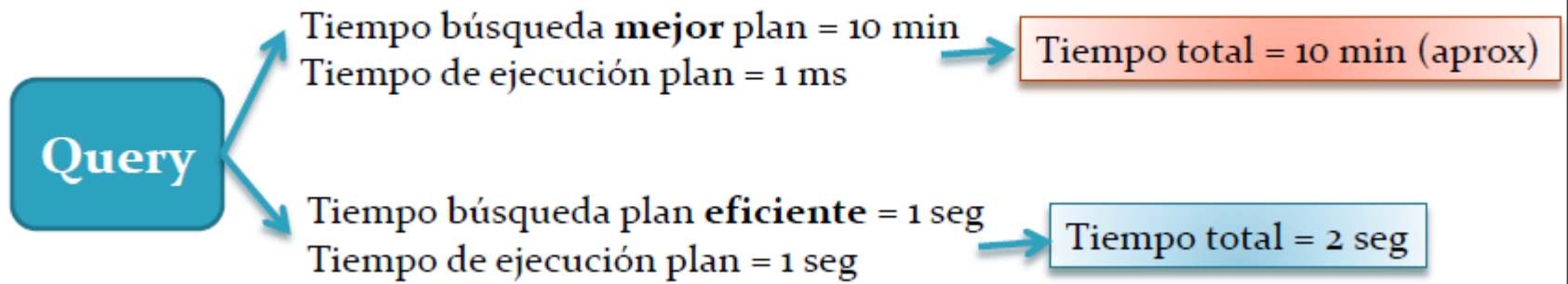
En cada bucket  **cantidad variable de bloques**



# Optimizador de consultas

Dada una consulta, encontrar un plan de ejecución eficiente

Plan de ejecución ➡ algoritmos y estructuras




Analizar todo el espacio de búsqueda ➡ costoso

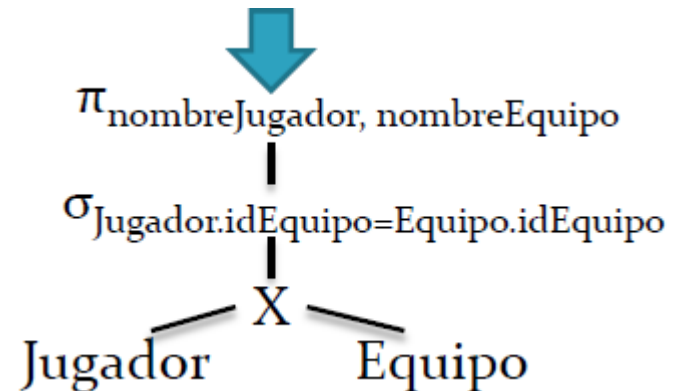
Evitar analizarlo completo, pero obtener planes eficientes

## Primer paso



```
SELECT nombreJugador, nombreEquipo  
FROM Jugador J, Equipo E  
WHERE J.idEquipo = E.idEquipo
```

  
 $\pi_{\text{nombreJugador, nombreEquipo}}(\sigma_{\text{Jugador.idEquipo=Equipo.idEquipo}}(\text{Jugador X Equipo}))$



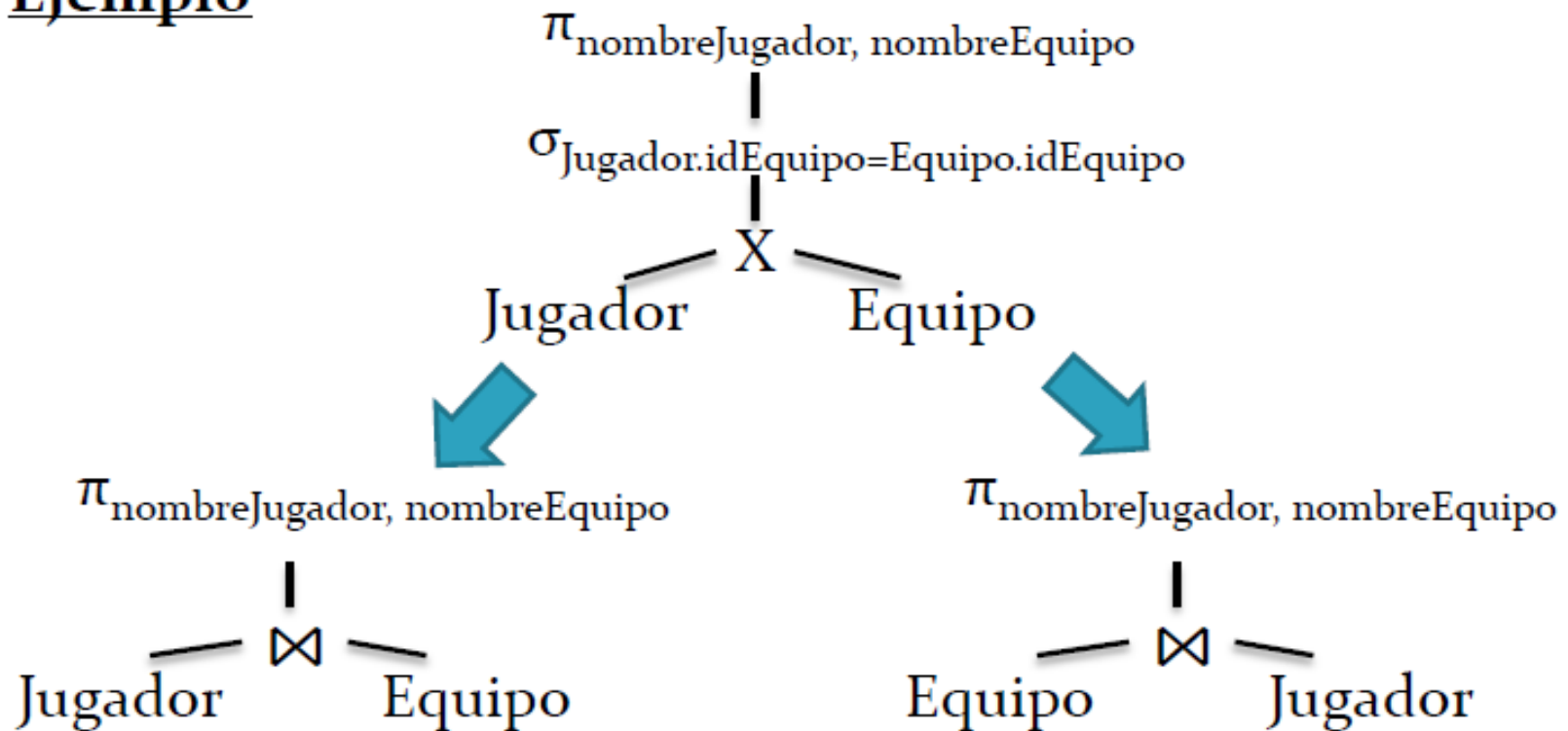
## Segundo paso

Modificaciones algebraicas sobre el árbol

Buscan mejorar la performance de la consulta **independientemente** de la organización física.

Involucran propiedades algebraicas que permiten construir una consulta **equivalente** a la original.

## Ejemplo





# Técnicas Algebraicas

# Heurísticas

Basadas en propiedades **algebraicas**

Árboles **equivalentes**

**Por lo general**, mejoran la performance de las consultas

# Ejemplos

Cascada de selecciones

Bajar selecciones

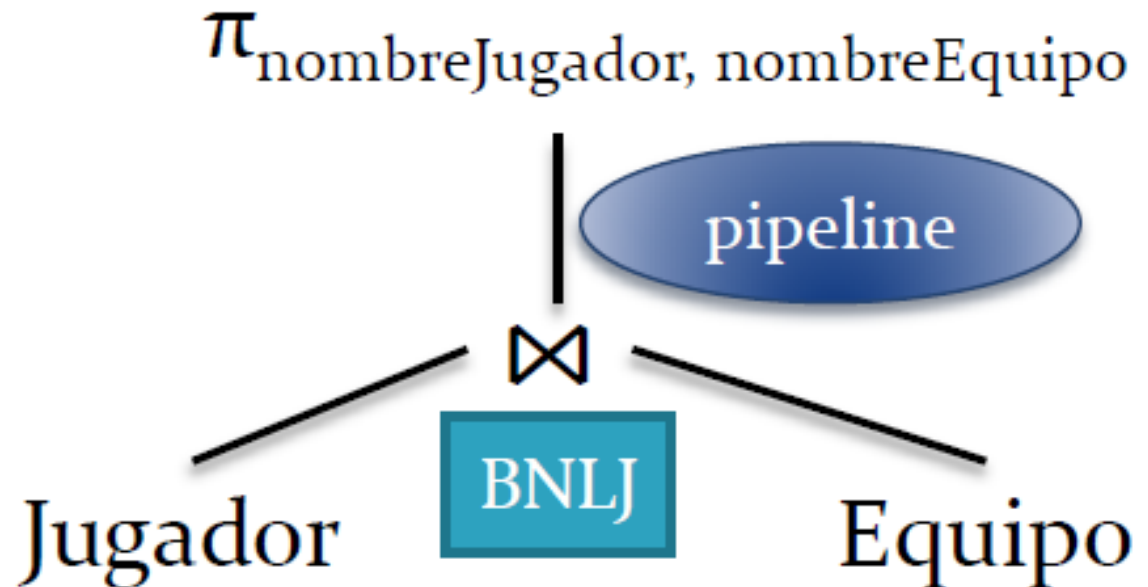
Bajar proyecciones

Cambiar productos cartesianos por joins

## Tercer paso

Selección de la implementación de cada operador

### Ejemplo



# Técnicas Físicas

Seleccionar implementaciones para los operadores basándose en **cómo** están organizados los archivos y las estructuras adicionales que existen

Utilizan el **Catálogo**

Información estadística de los datos

Se actualiza periódicamente y no está siempre sincronizado con los datos reales

Permite estimar la selectividad de los diferentes operadores

## ¿Cómo comparar planes?

- Se define un **modelo de costos**
- El costo será expresado en **cantidad de accesos a disco** (lecturas + escrituras)
  - Predomina sobre tiempo de CPU
- El costo de un plan será una estimación
- Se elegirá al plan con menor costo estimado

- Detalles:
  - No asumiremos nada sobre el *Buffer Manager*, por lo que siempre consideraremos que un pedido de lectura o escritura significa acceder a disco
  - En un bloque de R, hay sólo tuplas de R (y no de otras relaciones)
  - Las tuplas de R se guardan enteras en un bloque (por ejemplo, no puede haber mitad en el bloque i y la otra mitad en el bloque i+1)
  - Siempre asumiremos *peor caso*

¿Cómo se pasan los resultados entre nodos?

## Materialización

Los resultados intermedios se guardan **directamente en disco**.

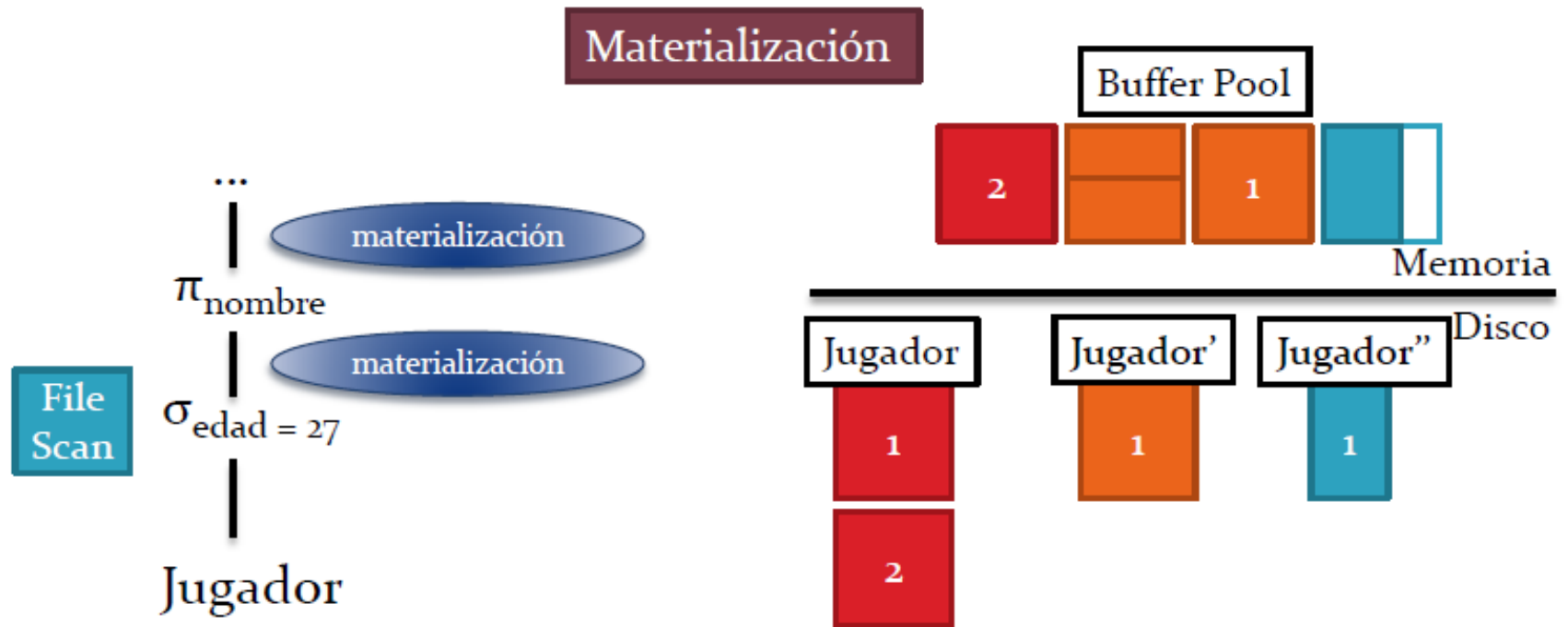
El siguiente nodo deberá levantarlos de disco **nuevamente**.

## Pipeline

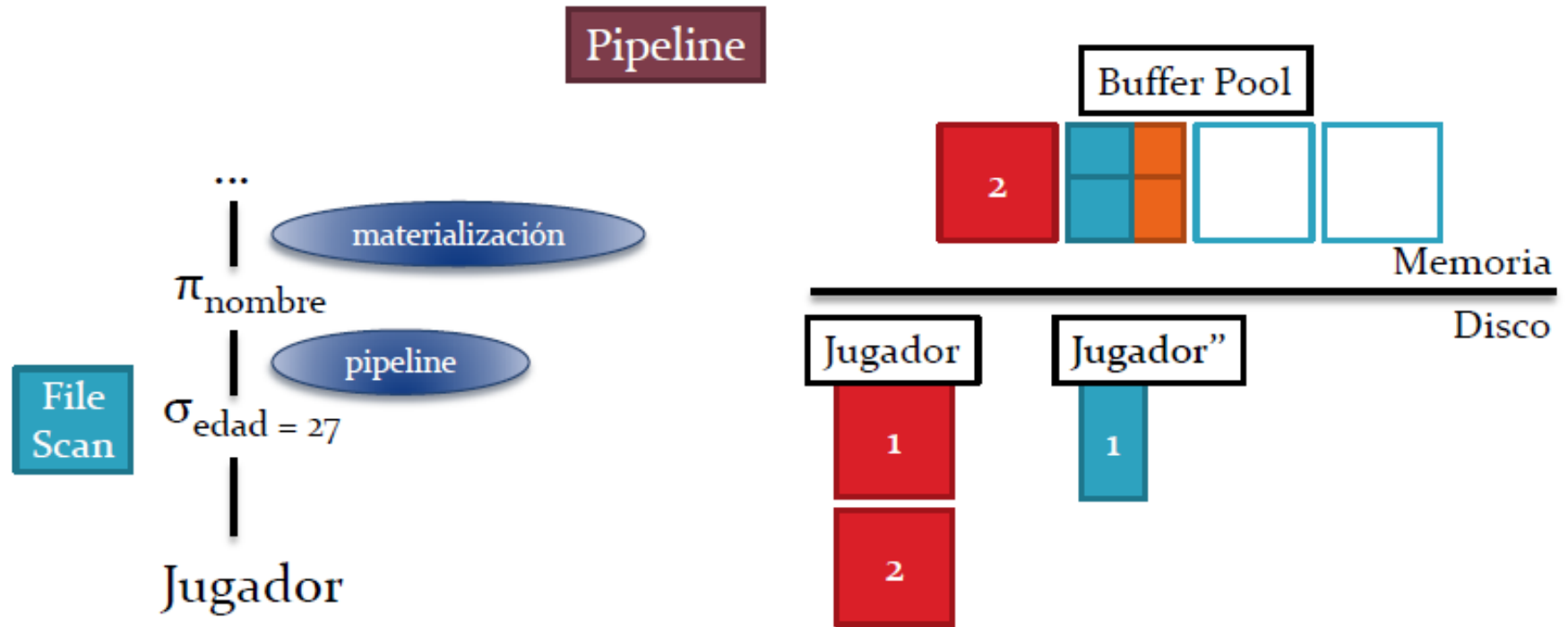
Las tuplas se van pasando al nodo superior **mientras** se continúa ejecutando la operación



# Materialización



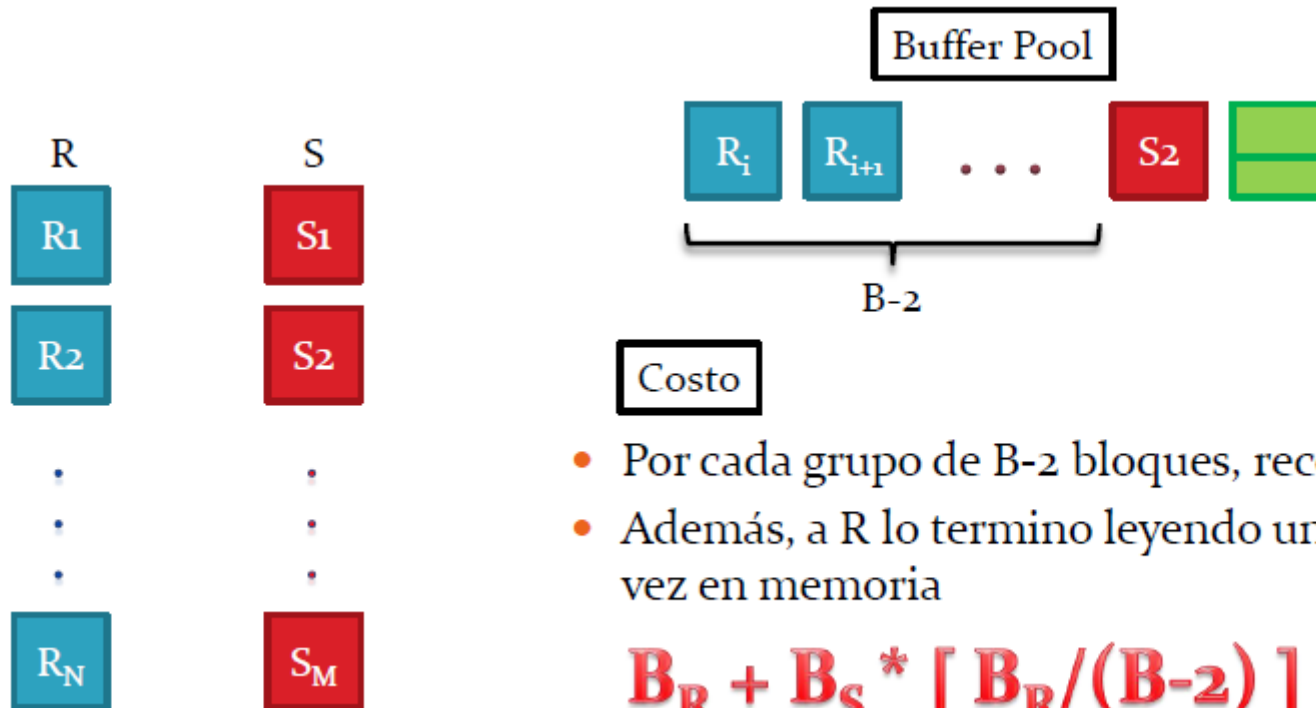
# Pipeline



**R∅S**

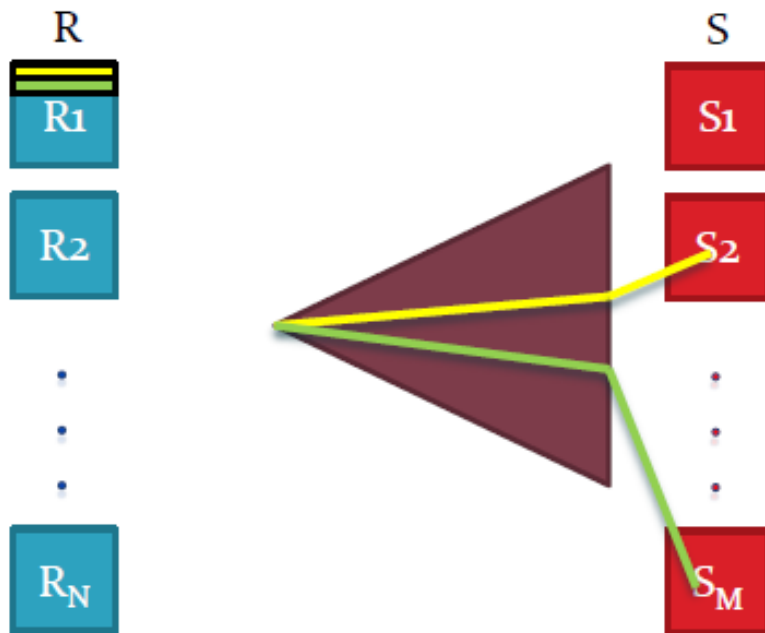
# Block Nested Loops Join (BNLJ)

B bloques de memoria



# Index Nested Loops Join (INLJ)

Índice I sobre S



Costo

- Por cada tupla de R, hago una búsqueda en el índice I
- Además, a R lo termino recorriendo una sola vez en memoria

$$B_R + T_R * \text{("costo índice")}$$



Dependerá del tipo de índice que exista

# Sort Merge Join (SMJ)

B bloques de memoria

| R    |     |
|------|-----|
| 1    | A   |
| 2    | B   |
| ...  | ... |
| 1000 | ABC |

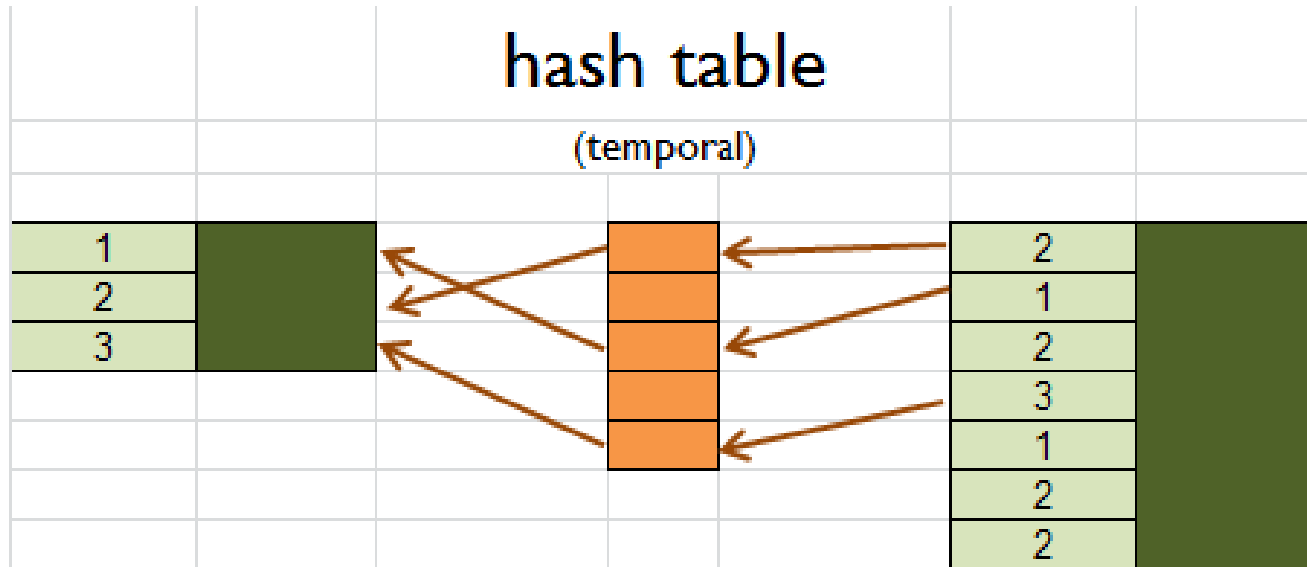
| S   |     |
|-----|-----|
| 1   | x   |
| 1   | y   |
| ... | ... |
| 100 | xyz |

Costo

- Se ordenan R y S (si alguno ya está ordenado, este costo no se considera)
- Se hace el *merge* entre R y S ordenados

$$(\lceil \log_{B-1} [B_R/B] \rceil + 1) * 2B_R + (\lceil \log_{B-1} [B_S/B] \rceil + 1) * 2B_S + B_R + B_S$$

# Hash Join (HJ)



Hash table debe entrar en  $B-1$  bloques de memoria  
The hash function divides the join attributes' domain into several ranges.  
The performance depends on the quality of the hash function and its implications, such as the number of buckets.

$$B_R + B_S$$