

Tema #1

Explique:

a- Medidas de rendimiento de los discos magnéticos.

- * Tiempo de Acceso: Tiempo que toma leer o escribir datos desde su petición hasta su confirmación.
- * Tiempo de Búsqueda: Tiempo que se tarda para ubicar el peine en la pista correcta varía de 4 a 10 ms.
- * Latencia Rotacional: Tiempo que tarda en aparecer el sector a escribir/ leer debajo de los cabezales. (de 4 a 15 ms)
- * Tasa de Transferecia: Volumen de Datos por unidad de tiempo que se pueden guardar o recuperar del disco.
 - * típicamente de 30 a 100 MB/s para lectura.
 - * 20 a 85 MB/s para escritura.
- * Tiempo Medio de Fallo (TMOF): lapso de tiempo que un disco puede trabajar continuamente sin fallas. la realidad de 3 a 5 años. En la teoría de 57 a 36 años.

b- Enfoques de organización de archivos de un SGBD.

- * Heaps: Los registros son guardados en cualquier lugar del archivo, en donde haya espacio suficiente. Los registros no se ordenan. Generalmente hay un archivo para cada relación.
- * Secuencial: Los registros se guardan en orden secuencial según el valor de la "clave de búsqueda" de cada uno.
- * Hash: Los registros son almacenados en un bloque especificado por una función hash que se aplica sobre el valor de un campo de registro.
- * Organización en agrupación: Los datos de diferentes relaciones se pueden guardar en el mismo archivo de registros.

c- Indices y cuando aplicarlos

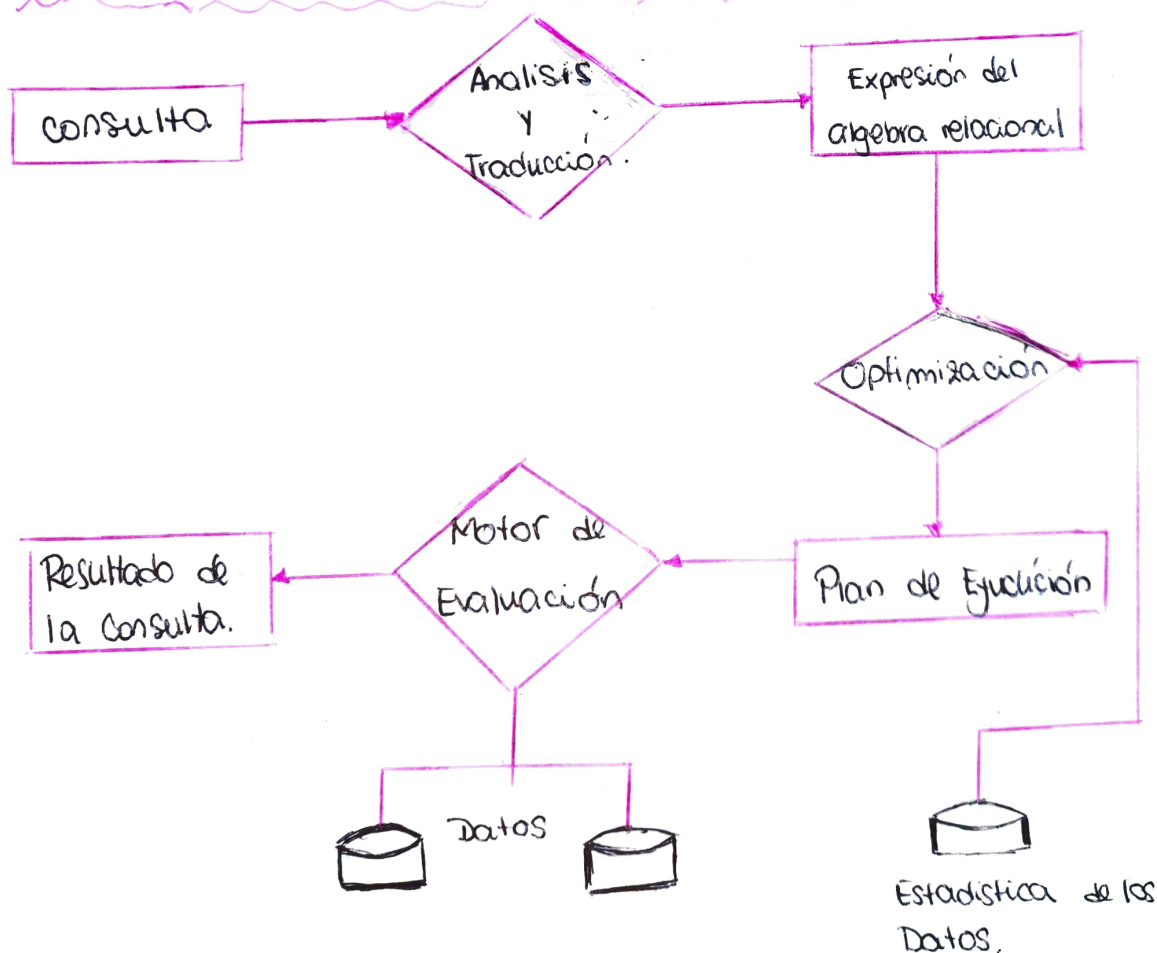
- * Indices Ordenados: Los registros índice están ordenados de acuerdo al orden de la clave de búsqueda. Pueden ser:
 - * Primarios: son aquellos cuya clave especifica el orden secuencial de archivo indexado.

* Secundarios: Son aquellos índices cuya clave de búsqueda no determinan el orden secuencial del archivo ordenado. Conviene usarlos cuando se tienen consultas que pidan rango de datos.

* Índices Asociativos: Son almacenados en registros (buckets) o bloques a partir de una función hash. Son adecuados cuando se piden datos que estén en condiciones de igualdad con la clave..

* Índices Bitmaps. Son un tipo especial de índices diseñado para la consulta sencilla sobre varias claves de manera eficiente, cada índice se construye para una única clave, por lo cual tiene un array de bits para cada valor del atributo del índice. Son útiles tanto para consultas sobre un atributo o sobre varios, siempre y cuando existe un índice Bitmap en cada atributo.

d- Pasos para el procesamiento de consultas. Diagrama.



Análisis y Traducción: Se revisa la sintaxis. Verifica en el catálogo del sistema que todos los identificadores sean nombres de objetos de la BD. Realiza la traducción a expresiones equivalentes en el álgebra relacional.

Fase de optimización: Minimiza el coste de la evaluación de la consulta, buscando la expresión de menor costo. Devuelve el plan de ejecución de la consulta como resultado en base a las estadísticas de datos.

Evaluación: Decide que tipo de evaluación se va a realizar a los datos si la materialización o encapsulamiento.

Tema #2

a-x Insertar 8, 9, 2

2 8 9

* Insertar 11, 7, 10

9

2 7 8 9 10 11

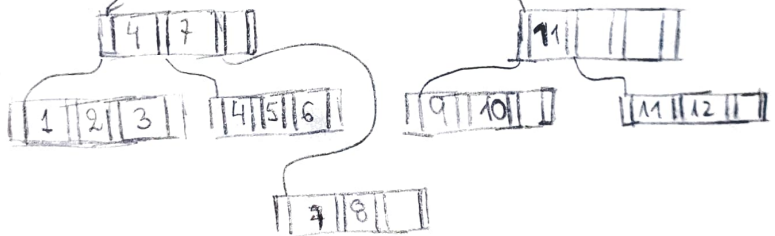
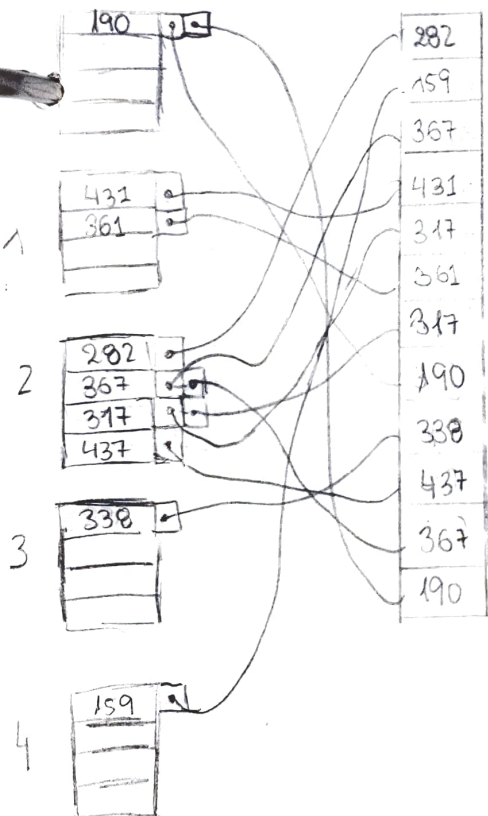
* Insertar 4, 5, 12

7 9 11
2 4 5 7 8 9 10 11 12

* Insertar 1, 3, 6

9

b) Cajones de 4 elementos



c) Para el árbol B+ el índice es primario, en los nodos hijos los registros índices apuntan a los registros de la tabla, pues en este caso el registro índice es el ID y mantiene ordenada la tabla

• Para el índice hash es secundario, para el par (puntero, clave) apunta a los cajones donde se encuentran los registros, ésta no está construida sobre ninguna clave.

b) cajones de 3 elementos.



Tema #3

$$r_1: n_1 = 40000 \text{ tuplas} ; f_1 = 100 \text{ t/b} ; b_1 = 400$$

$$r_2: n_2 = 90000 \text{ tuplas} ; f_2 = 45 \text{ t/b} ; b_2 = 2000$$

"r1 join r2 on (r1.C = r2.C)"

a- Reunión de bucle anidado

$$\text{Costo cant. Accesos} = n_1 b_2 + b_1 = 40000 \cdot 2000 + 400 = 80000400 \text{ accesos por bloque}$$

b- Reunión en bucle anidado por bloques

$$\text{Costo cant. Accesos} = b_1 b_2 + b_1 = 400 \cdot 2000 + 400 = 800400 \text{ accesos por bloque}$$

c- Reunión por mezcla

r_1 ordenada externamente $M=3$

$$\begin{aligned} \text{Costo ordenamiento} &= b_1 \cdot (2 \log_{M-1} \left(\frac{b_1}{M} \right) + 1) = 400 \cdot (2 \log_2 \left(\frac{400}{3} \right) + 1) \\ &= 400 \cdot (2 \cdot 7 + 1) \\ &= 6000 \end{aligned}$$

$$\text{Costo de cant. acceso} = b_1 + b_2 + C = 400 + 2000 + 6000 = 8400 \text{ accesos por bloque}$$

d- Reunión de bucle anidado indexada

r_2 60 punteros

$$\text{Costo} = b_1 + n_1 (\log_{P/2} n_2 + 1) = 2000 + 40000 (\log_{30} 90000 + 1)$$

Tema #4

* Búsqueda Lineal: Cuando no se tiene el índice de clave de búsqueda ni se tiene ordenada la tabla. Necesariamente se revisan todos los bloques.

$$\text{Costo} = br \text{ (cant. de bloques)}$$

* Búsqueda Binaria: Cuando no se tiene un índice construido sobre el atributo de búsqueda pero la tabla está ordenada por el atributo.

$$CA_2 = \lceil \log_2 (br) \rceil + \lceil CS(A, r) / fr \rceil - 1$$

* Búsqueda en índice primario para un atributo clave
cuando se tiene un índice de clave de búsqueda que además es un atributo clave.

$$\text{Costo} = AA_i + 1 \rightarrow \text{lectura del bloque}$$

↳ Recorrido del Arbol.

* Búsqueda en índice. ^{Primario.} ~~Secundario~~ para cada atributo no clave.

El costo está determinado por la cantidad de bloques que contienen registros que cumplen con la condición de igualdad más la altura del índice.

$$\text{Costo} = C_{AI} = AA_i + [CS(A_i) / f_i]$$

- Índice secundario: Cuando no se tiene ningún índice de clave de búsqueda ni tampoco algún atributo clave.

$$\text{Costo} = b_r + C(A_i, r) + 1 = AA_i + 1 + CS(A_i)$$

Tema #5

select distinct A.a1, C.c1 from A join B on (A.a2 = B.b3) join C on (C.c2 = B.b4) where A.a1 > 10 and B.b1 = 50;

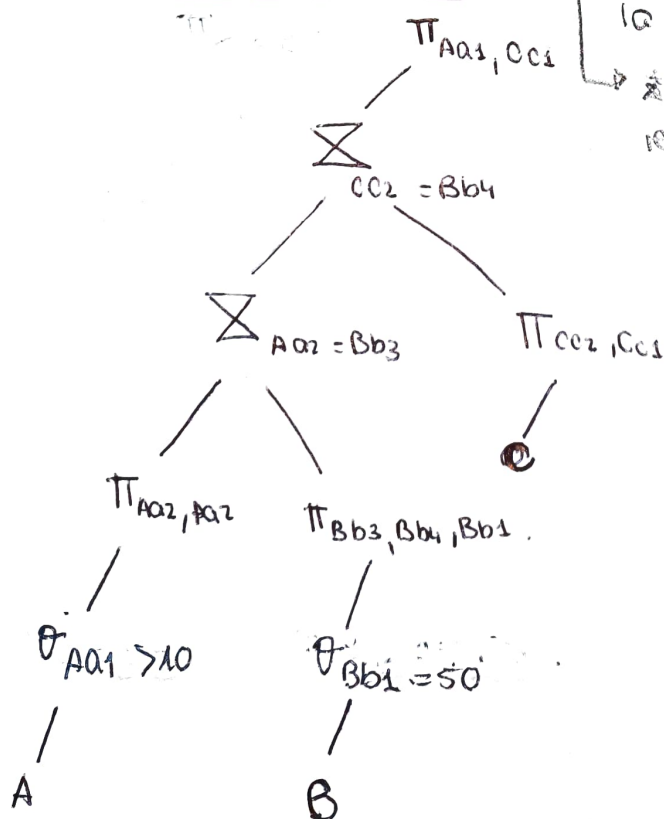
Algebra Relacional.

a) $\pi_{A.a1, C.c1} \left(\left(\left(\theta_{A.a1 > 10} \wedge B.b1 = 50 \right) \left(A \bowtie_{A.a2 = B.b3} B \right) \bowtie_{C.c2 = B.b4} \right) \right)$

b) Realizar primero los select y dejar para el último los join.

$$\pi_{A.a1, C.c1} \left[\left(\pi_{A.a1, A.a2} \left(\theta_{A.a1 > 10} (A) \right) \bowtie_{A.a2 = B.b3} \pi_{B.b3, B.b1, B.b4} \left(\theta_{B.b1 = 50} (B) \right) \right) \bowtie_{C.c2 = B.b4} \pi_{C.c2, C.c1} (C) \right]$$

c) Árbol de la expresión final.



→ se aplica la regla 7, en la cual se distribuye sobre la operación de join.
→ también se utiliza la regla 8, en la cual la proyección sobre los joins.

Tema # : Determine si las siguientes planificaciones son o no secuenciadas en cuanto a vistas e indique el orden de las transacciones en caso afirmativo

Plan 1	
T ₁	T ₂
Leer (A)	Escribir (A)
Escribir (A)	Escribir (A)

Si



Plan 1	
T ₁	T ₂
Leer (A) Escribir (A)	Escribir (A) Escribir (A)

Plan 2	
T ₁	T ₂
Leer (A)	Escribir (B)
Leer (B)	Escribir (A)

NO

Plan 3		
T ₁	T ₂	T ₃
Leer (A)	Leer (B)	Escribir (B)
Escribir (A)	Escribir (B)	Leer (A)

Si



Plan 3		
T ₁	T ₂	T ₃
Leer (A) Escribir (A)	Leer (B) Escribir (B)	Escribir (B) Leer (A)

Tema II : Determine si las siguientes planificaciones son o no secuenciables en cuanto a conflictos e indique cual es el orden de las transacciones en caso afirmativo.

Plan 1	
T ₁	T ₂
leer (A)	Escribir (A)
Escribir (A)	
	Escribir (A)

- NO

Plan 2	
T ₁	T ₂
leer (A)	Escribir (B)
leer (B)	Escribir (B)

- NO

Plan 3		
T ₁	T ₂	T ₃
	Escribir (B)	
Escribir (A)		Escribir (C)
	leer (C)	

Si

Plan 3		
T ₁	T ₂	T ₃
Escribir (A)		
	Escribir (B)	
	leer (C)	
		Escribir (C)



Tema # :

Tema #1 : Sitio 1

$R \text{ join } S \text{ on } (R.A2 = S.A2)$

$R @ S_1$	
A_1	A_2
1	3
1	4
1	6
2	3
2	6
3	7
3	8
3	9

$S @ S_2$		
A_2	A_3	A_4
3	13	16
3	14	16
7	13	17
10	14	16
10	15	17
11	15	16
11	15	16
12	15	16

Paso #1 : calcular temp1

$\text{temp1} = \Pi_{R.A2} (A_2) \text{ en } S_1$

$\text{temp1} = \begin{bmatrix} 3 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$

Paso #2 : Transferir temp1 a S_2

Costo $P_2 = 6 \text{ ui}$

Paso #3 : calcular $\text{temp2} = S \Join \text{temp1}$

$\text{temp2} = \begin{bmatrix} A_2 & A_3 & A_4 \\ 3 & 13 & 16 \\ 3 & 14 & 16 \\ 7 & 13 & 17 \end{bmatrix}$

Paso #4 : Transferir temp2 a S_1

Costo $P_4 = 9 \text{ ui}$

Paso #5 : Dar el Resultado Final

$\text{Res} = R \Join \text{temp2}$

$\text{Res} = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ 1 & 3 & 13 & 16 \\ 1 & 3 & 14 & 16 \\ 2 & 3 & 13 & 16 \\ 2 & 3 & 14 & 16 \\ 3 & 7 & 13 & 17 \end{bmatrix}$

Costo = Costo P_2 + Costo $P_4 = 6 \text{ ui} + 9 \text{ ui} = 15 \text{ ui}$

Tema

Estrategia más simple

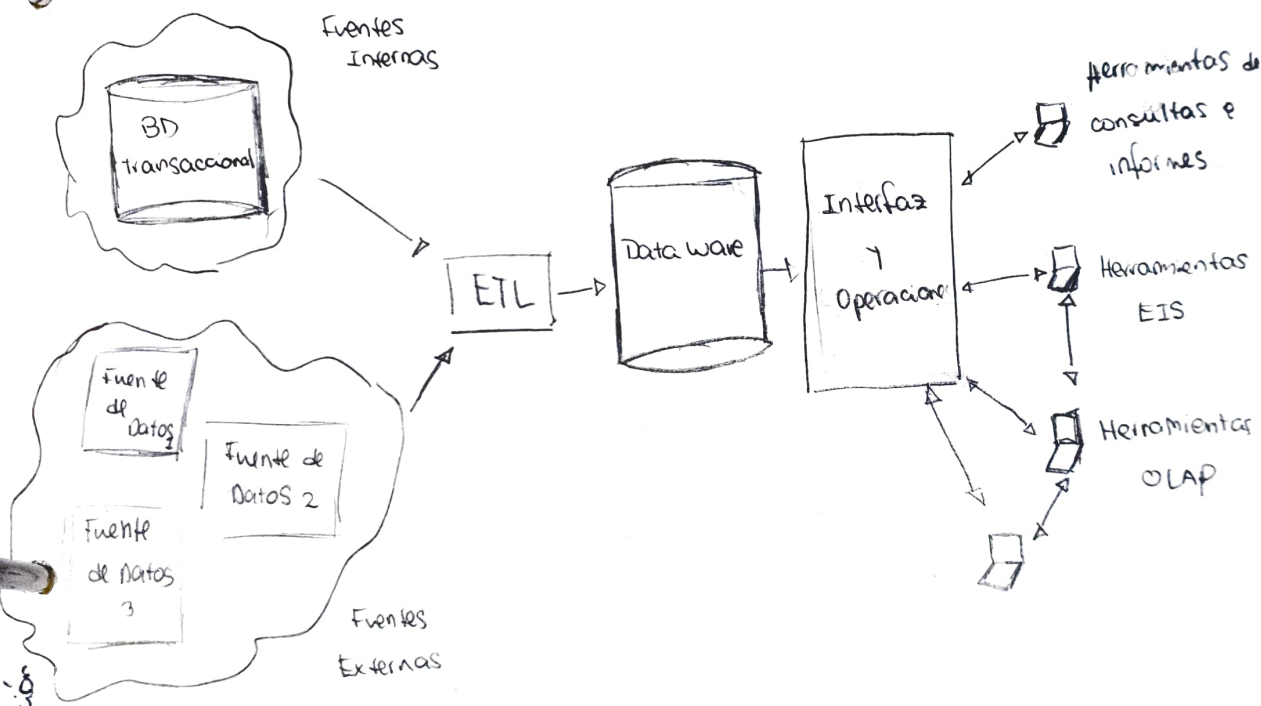
Paso #1: Transferir S a S_1

Paso #2: consultar el Resultado $Res = R \bowtie S'$ (S' : Resultado $S \rightarrow S_1$)

Costo: Tamaño del $S_2 = 3 \times 8 = 24$ unidades

- * En la semi Reunión se ahorra más en los datos transferidos por ello tiene mejor costo que la estrategia más simple

Tema # : Arquitectura de DataWare



* ETL: se encarga de la Extracción de las fuentes de Datos, la transformación y envío de datos al DW.

* Dataware: Es la Base de datos que se utiliza para la toma de decisiones y construida a partir de los registros de las transacciones del negocio.

* OLAP: Herramienta que realiza el Análisis en línea de manera rápida de las informaciones.

Tema #

Tema # : Sitio 2

R join S on (R.A2 = S.A2)

R@S1	
A1	A2
1	3
1	4
1	6
2	3
2	6
3	7
3	8
3	9

S@S2		
A2	A3	A4
3	13	16
3	14	16
7	13	17
10	14	16
10	15	17
11	15	16
11	15	16
12	15	16

Paso #1 : calcular temp1

temp1 = $\Pi_{RDS}(A2)$ en S2

temp1 = $\begin{bmatrix} 3 \\ 7 \\ 10 \\ 11 \\ 12 \end{bmatrix}$

Paso #2 : Transferir temp1 a S1

costo Paso2 = 5 ui

Paso #3 : calcular temp2 = R \bowtie temp1

temp2 = $\begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 3 & 7 \end{bmatrix}$

Paso #4 : Transferir temp2 a S2

costo Paso4 = 6 ui

Paso #5 : Resultado Res = S \bowtie temp2

Res = $\begin{bmatrix} 3 & 7 & 7 & 13 & 17 \\ 1 & 3 & 3 & 13 & 16 \\ 1 & 3 & 3 & 14 & 16 \\ 2 & 3 & 3 & 13 & 16 \\ 2 & 3 & 3 & 14 & 16 \end{bmatrix}$

Costo = costo Paso2 + costo Paso4

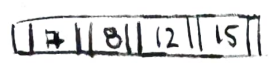
Costo = 5 ui + 6 ui

Costo = 11 ui

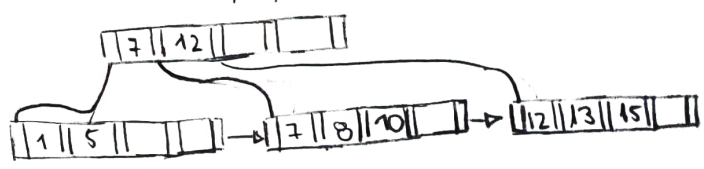
Tema #

a) Orden : 15, 8, 7, 12, 13, 5, 10, 1, 14, 3, 11, 2, 4, 6, 9

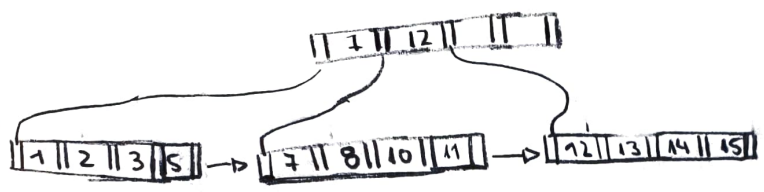
* Insertar : 15, 8, 7, 12



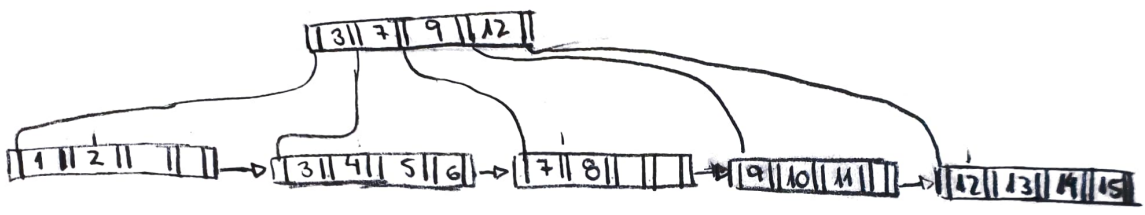
* Insertar : 13, 5, 10, 1



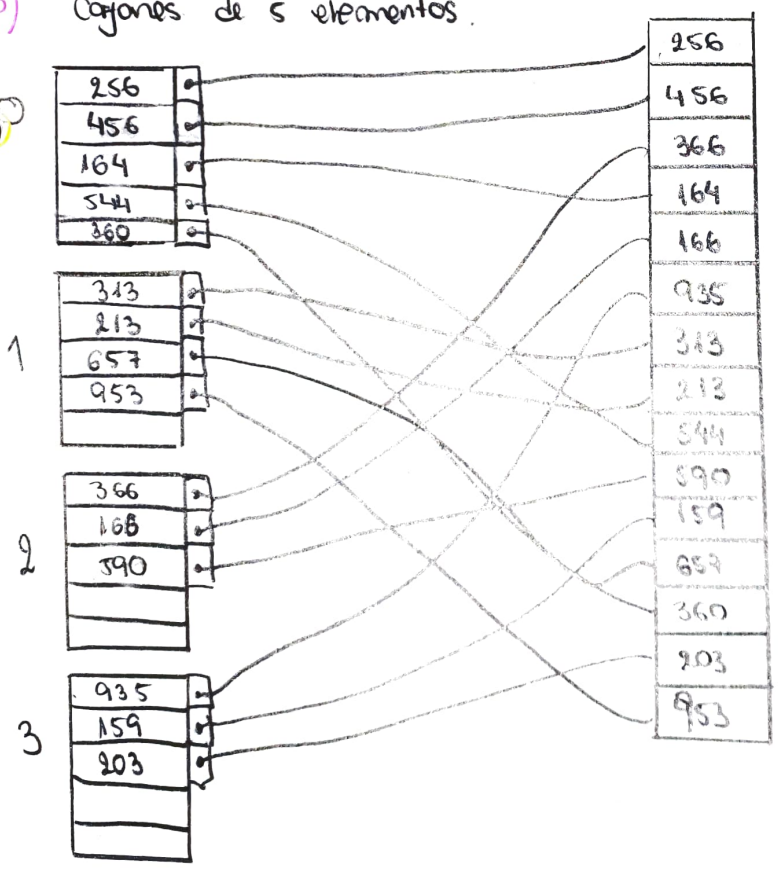
* Insertar : 14, 3, 11, 2



* Insertar : 4, 6, 9



b) Conjuntos de 5 elementos.



c) * Para el árbol B+ el índice es primario, en los nodos hijos los registros índices apuntan a los registros de la tabla, pues en este caso el registro índice es el ID y mantiene la la tabla ordenada.

* Para el índice Hash, el índice es secundario, para el par (puntero, clave) apunta a los conjuntos donde se encuentran los registros, ésta no está construida sobre ninguna clave.

Tema 4 :

$$r_1: m = 35000 \text{ tuplas} ; f_1 = 50 \text{ tuplas/bloque} ; b_1 = \frac{35000}{50} = 700$$

$$r_2: n_2 = 75000 \text{ tuplas} ; f_2 = 100 \text{ tuplas/bloque} ; b_2 = \frac{75000}{100} = 750$$

" $r_1 \text{ join } r_2 \text{ on } (r_1.C = r_2.C)$ "

a) Reunión por mezcla

r_1 ordenada externamente $M=3$

$$\begin{aligned} \text{costo} &= b_1 + b_2 + b_1 (2 \lceil \log_{M-1} \left(\frac{b_1}{M} \right) \rceil + 1) \\ &= 700 + 750 + 700 (2 \lceil \log_2 \left(\frac{700}{3} \right) \rceil + 1) \\ &= 1450 + 700 (2 \cdot 8 + 1) \\ &= 13350 \text{ accesos p/ bloque} \end{aligned}$$

b) Reunión por bucket anidado indexado

r_2 con $p = 60$ punteros

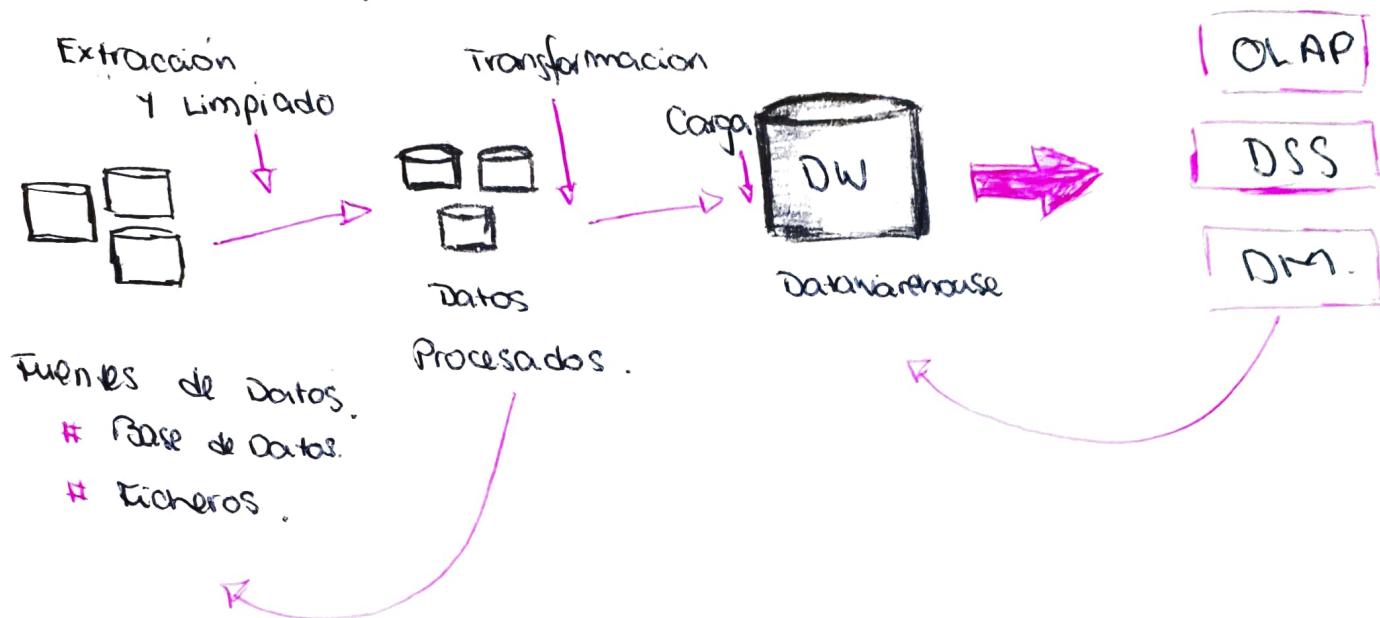
$$\begin{aligned} \text{costo} &= b_1 + n_1 (\lceil \log_{p/2} n_2 \rceil + 1) \\ &= 700 + 35000 (\lceil \log_{30} 75000 \rceil + 1) \\ &= 700 + 35000 (4 + 1) \\ &= 175700 \text{ accesos p/ bloque} \end{aligned}$$

c) Reunión por Hash

$$M = 20 \rightarrow N = \frac{b_1}{M} = \frac{700}{20} = 35 \quad N > M$$

$$\begin{aligned} \text{costo} &= 2(b_1 + b_2) \lceil \log_{M-1} (b_2 - 1) \rceil + (B_r + B_s) \\ &= 2(700 + 750) \lceil \log_{19} 750 - 1 \rceil + 700 + 750 \\ &= 2 \times 1450 \times 2 + 700 + 750 \\ &= 4352 \text{ accesos p/ bloque} \end{aligned}$$

Tema # : Proceso de Extracción, transformación y carga (ETL) de datos en DW



- Extracción: Primer paso para obtener la información hacia el ambiente del Data Warehouse.
- Transformación: Una vez que la información es extraída hacia el área de tráfico de datos, algunas tareas de esta fase:
 - limpieza de información
 - Selección de los campos necesarios para el DW
 - combinación de fuentes de Datos.
- Carga: Final del proceso de transformación, los datos están en forma para ser cargados.