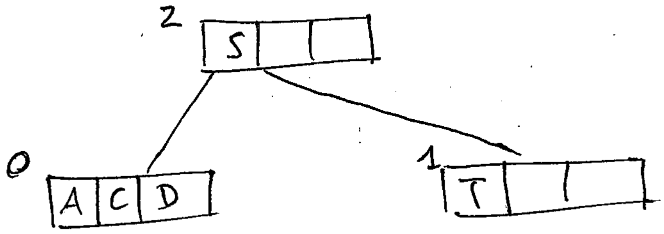
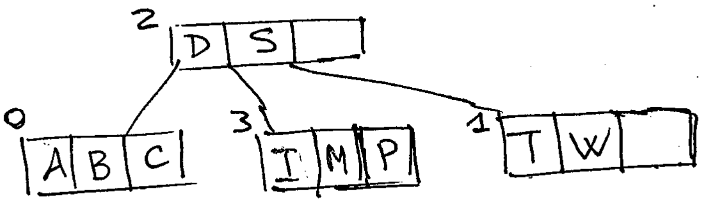
Ejemplo: K=3 , insertar C S D T A M P I B W N G U R K E H O L J Y Q Z F X V

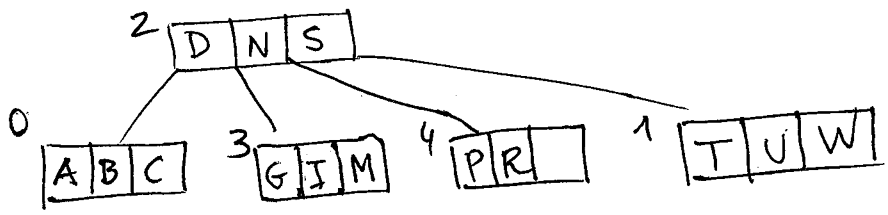
insertamos T:



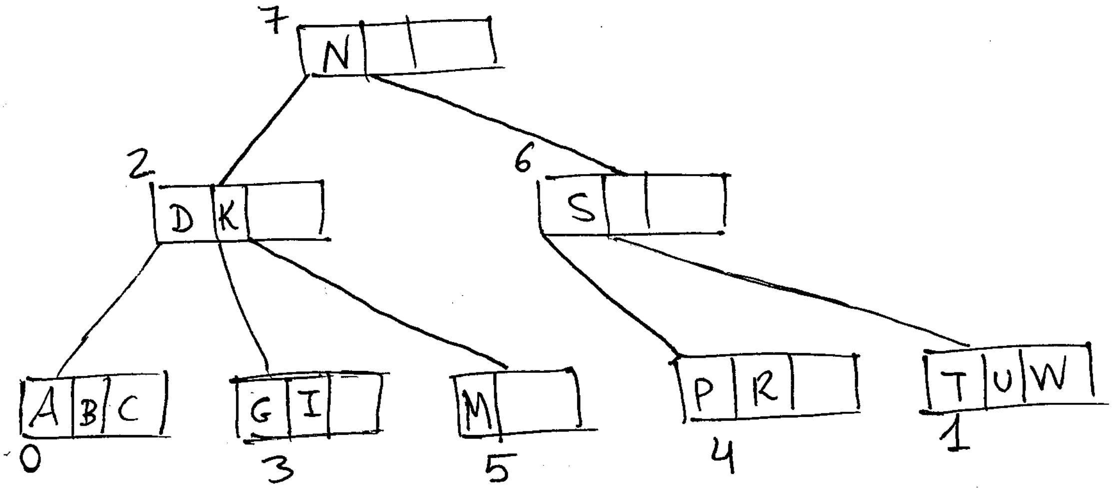
insertamos M:



insertamos N:



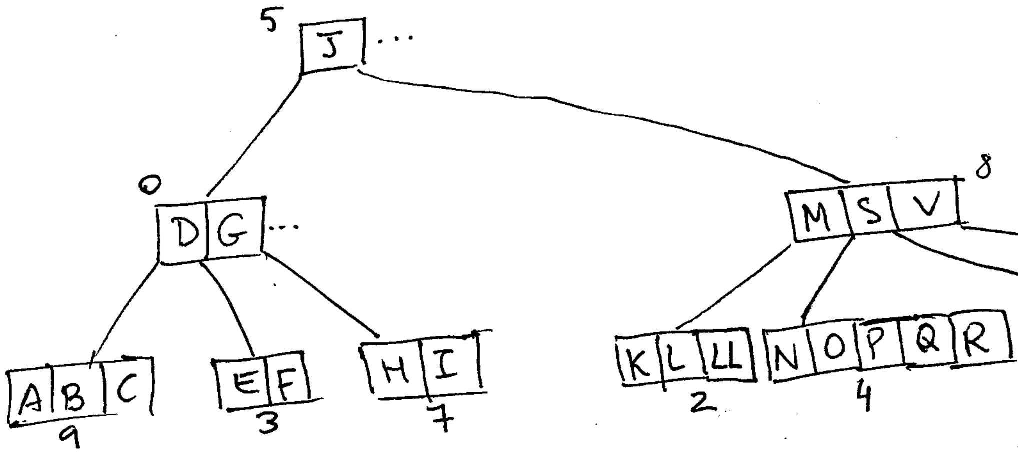
insertamos K:



Número mínimo de claves por nodo = $\lfloor K/2 \rfloor$.
Excepción: la raíz

[...]

Ejemplo (eliminación de claves) K = 5

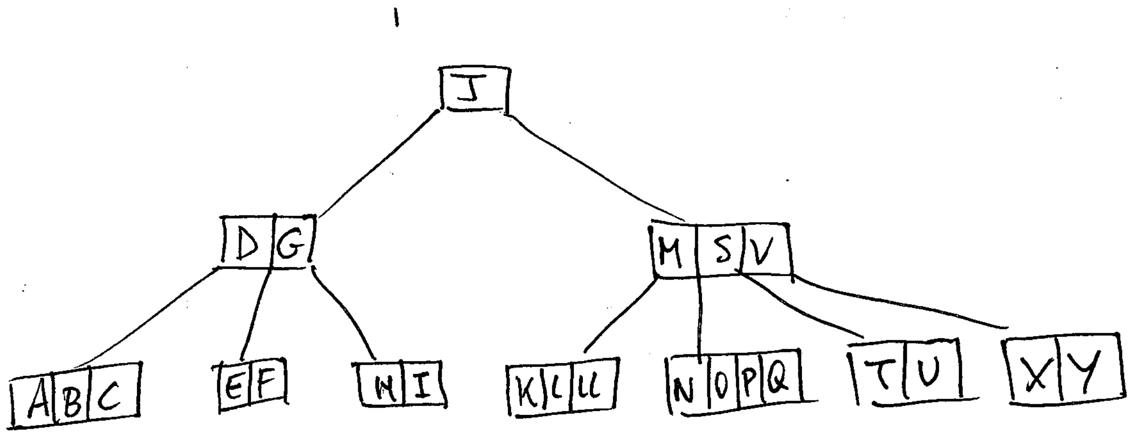


Eliminar: R, J, E, H, M

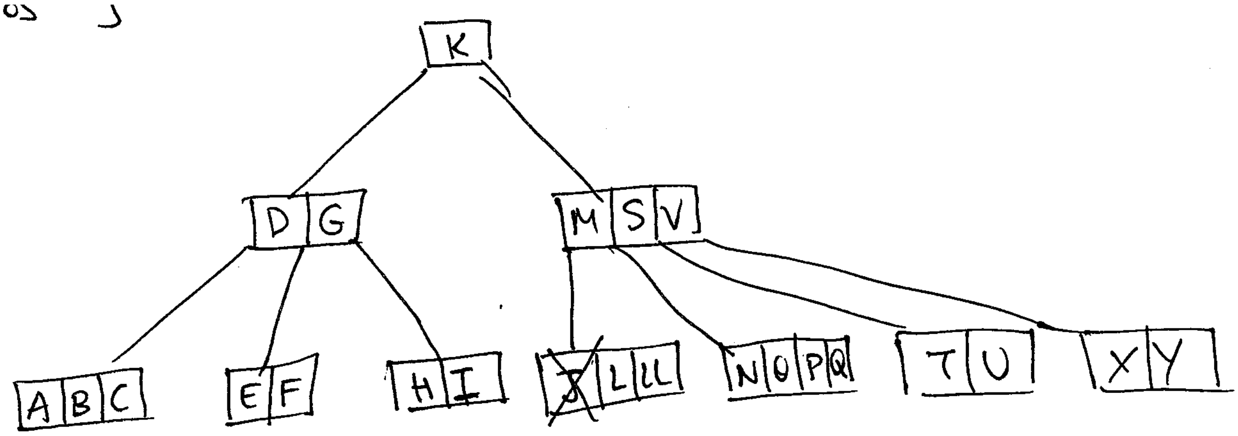
Algoritmo:

1. Buscar clave
2. Si no es una hoja, intercambiamos con la clave anterior o poste de una hoj
3. Eliminar la clave de la hoja
4. Si la página queda con menos de $\lfloor \frac{K}{2} \rfloor$ clave
   a. Si a una página hermana consecutiva le sobran claves, redistribuimos.
   b. En otro caso, concatenar (baja clave + merge)

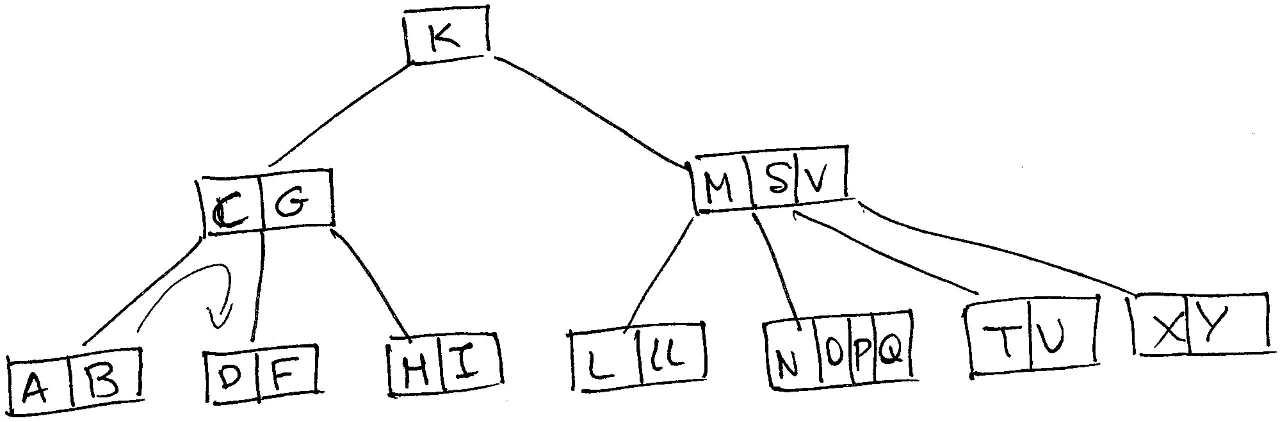
Eliminamos R



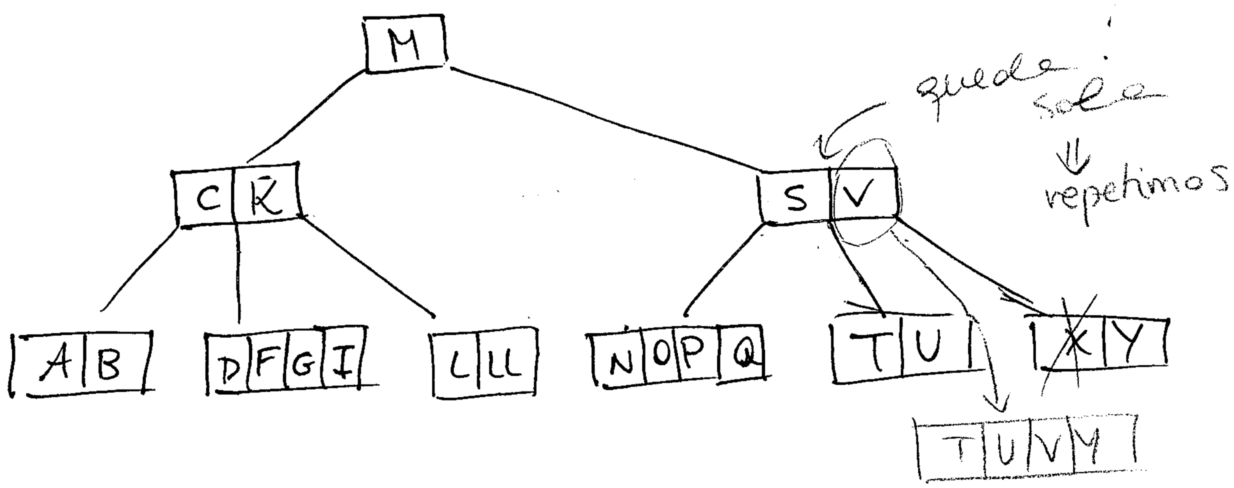
Eliminamos J



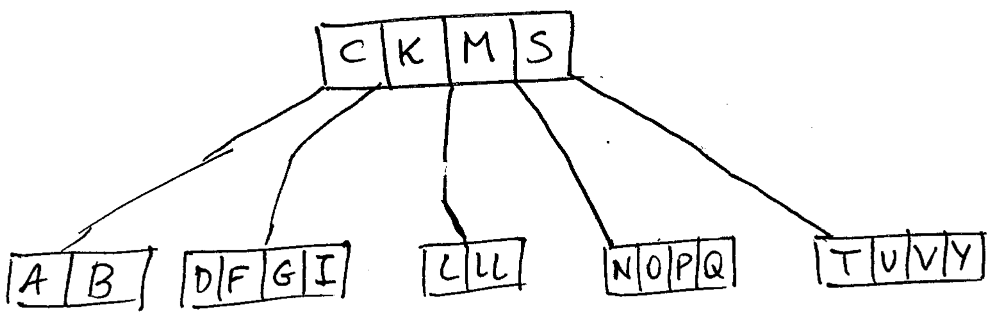
Eliminamos E



Eliminamos H:

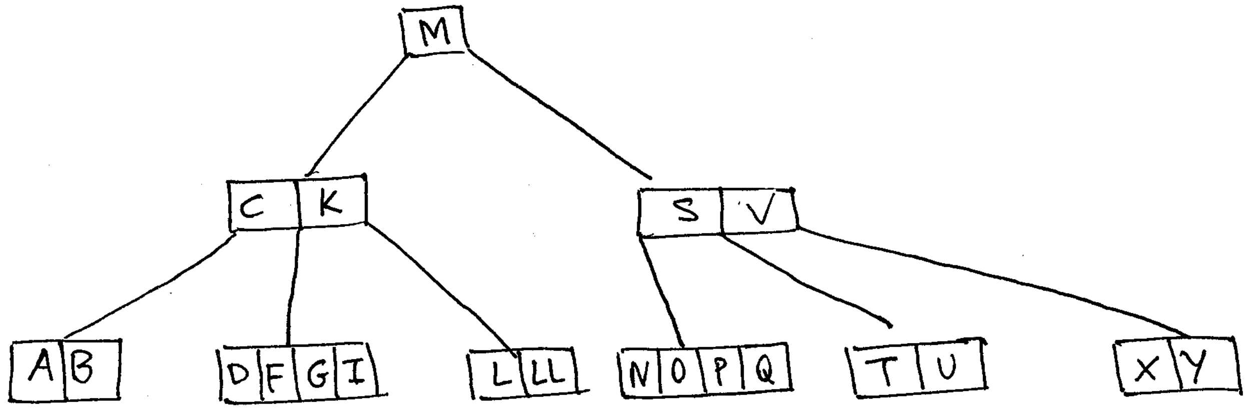


Eliminamos X:



Eliminamos M:

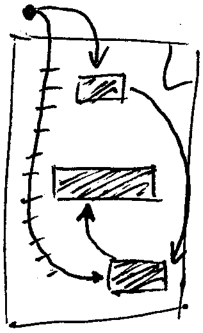
ANTES



DESPUÉS

Mínimo llenado Árbol B+ $\lceil \frac{K}{2} \rceil$, pero en la parte superior (lo que no son bloques, sino un árbol B como tal) se sigue cumpliendo que el mínimo llenado es $\lfloor \frac{K}{2} \rfloor$.

1. LONGITUD FIJA



insertamos en el primero

2. LONGITUD VARIABLE

insertamos en el primero que quepa

insertar en el primero
- First fit: no ordenar
- Best fit: ordenar de menor a mayor
- Worst fit: ordenar de mayor a menor

$R(a, b, c, d, e)$

$\{b, c\} \to \{a, d, e\}$

$d \to b$

clave $\{b, c\}$
clave $\{d, c\}$

$d \to b$ ~~BCNF~~ 3NF ✓

$\{b, c\} \to a$
$\{b, c\} \to d$
$\{b, c\} \to e$
BCNF ✓

EJERCICIO PARCIAL 1 NORMAL FORMS

$A \to \{B, C, D, E, F\}$

$\{B, C\} \to \{A, D, E, F\}$

$D \to E$

claves: A
$\{B, C\}$

1. Dependencias mínimas

$A \to B$ $A \to C$ $A \to D$ $A \to E$ $A \to F$
$\{B, C\} \to A$ $\{B, C\} \to D$ $\{B, C\} \to E$ $\{B, C\} \to F$
$D \to E$

2. las juntamos con la misma parte izda.

$A \to \{B, C, D, E, F\}$ $\{B, C\} \to \{A, D, E, F\}$

$R_1(A, B, C, D, E, F)$
$R_2(B, C, A, D, E, F)$
redundantes 3.
$R_3(D, E)$

$R_1(\underline{A}, B, C, D, E, F)$
$R_3(\underline{D}, E)$
$\to R_1(\underline{A}, B, C, D, F)$
$R_3(\underline{D}, E)$

esto se puede ver antes

$A \to \{B, C, D, E, F\}$ BCNF

$\{B, C\} \to \{A, D, E, F\}$ BCNF

$D \to E$ ~~BCNF~~ ~~3NF~~ ~~1NF~~ 2NF

15.

$5 \cdot 10^6$ registros de 400 B

Disco:

seek promedio 8 ms

rotación 15000 rpm

Sector 1000 B

sectores/pista: 500

sectores/bloque: 4

↳ 1 bloque = 4000 B

↳ 1 pista = 500 000 B

(*) 1 pista se lee en 1 vuelta (4ms)

(a)
```
while not eof
    fread ( ) //un registro
    hacer algo con el reg
```

(c)
```
fread (registros) -> leemos tod
for r in registros // RAM
    hacer algo con reg
```

(b)
```
while not eof
    fread (buffer) //400000 B
    for r in buffer
        hacer algo con reg.
```

a) ii) Por cada registro

→ seek 8 ms

→ latencia $= \dfrac{\text{tiempo 1 vuelta (ms)}}{2} = \dfrac{\frac{60000 \cdot 1}{15000}}{2} = 2$ r

→ transferencia $= \dfrac{4000\text{ B}}{500\,000\text{ B}} \cdot 4\text{ ms} = 0'032\text{ ms}$ regla de tres (*)

la suma se mult. por nº regs $\Rightarrow 10'032\text{ ms} \cdot 5 \cdot 10^6 \text{ regs} = 50160\text{ s}.$

b) ii) Por cada vez que se llena el buffer

1 buffer = 100 bloques

→ seek 8 ms

→ latencia 2 ms

] (×100) fragmentación

→ transferencia $= \dfrac{400000 \text{ buffer}}{500000 \text{ pista}} \cdot 4\text{ ms} = 3'2\text{ ms}$ (*) regla de tres

total = 1003'2 ms por buffer

$\Rightarrow \dfrac{5 \cdot 10^6 \cdot 400\text{ B}}{400\,000\text{ B}}$ = nº vez que hay que llenar el buffe

b) i) sin multiplicar por 100

$\Rightarrow 1003'2\text{ ms} \cdot 5000 = 501'6\text{ s}.$

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{150}$

PM U

P
U

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{250}$

PM U

P
U

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{350}$

PM U

U P

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{450}$

PM U

P
U

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{50}$

PM U

U P

100	200	300	400	500
0	1	2	3	4
P		M		U

$\boxed{1000}$

PM U

P
U

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{150}$

PM U

P
U

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{250}$

PM U

P
U

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{350}$

P M U

P
U

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{450}$

P M U

P
U

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{50}$

PM U

U P

100	200	300	400	500	600
0	1	2	3	4	5
P		M			U

$\boxed{1000}$

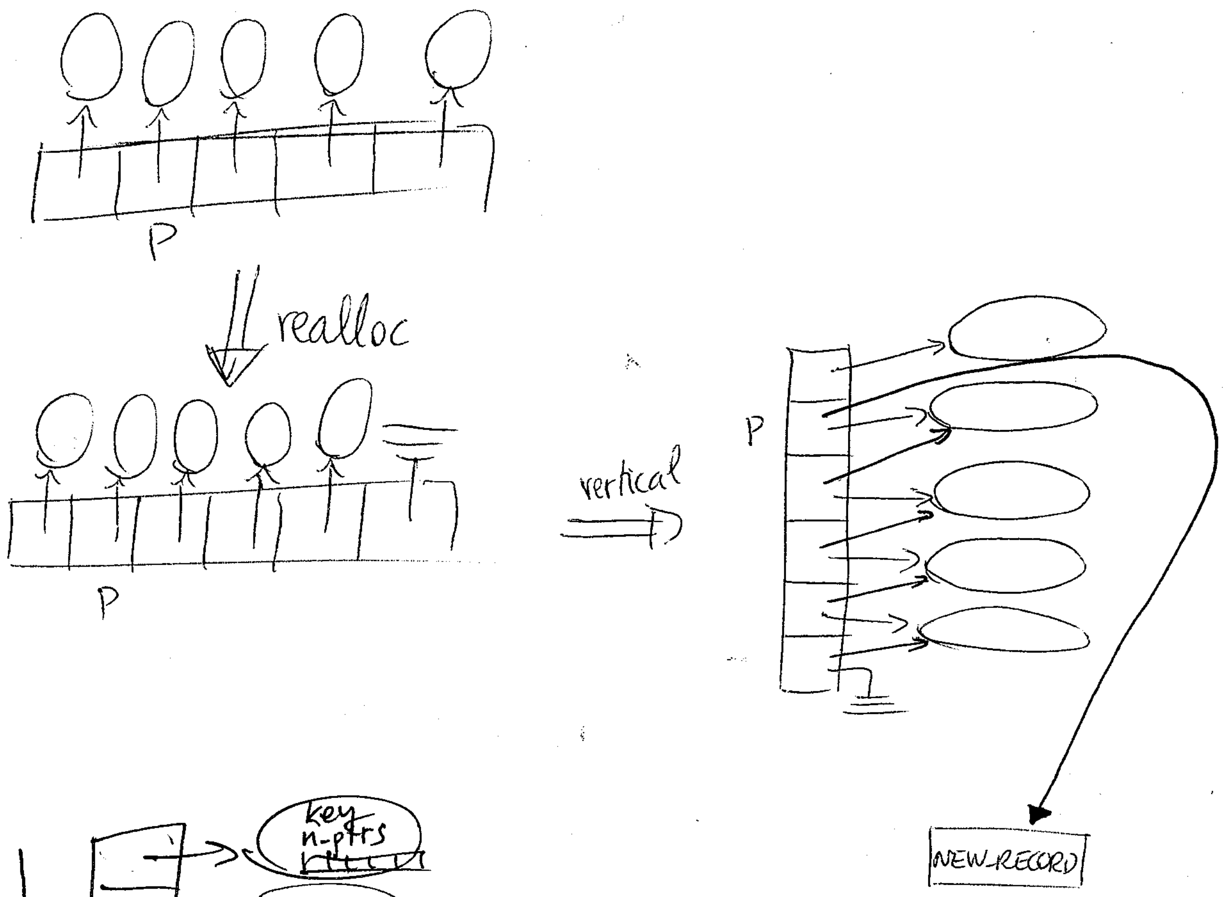
P M U

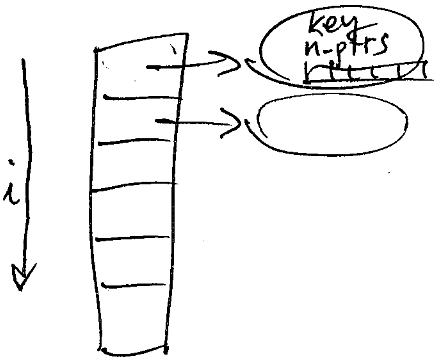
P
U

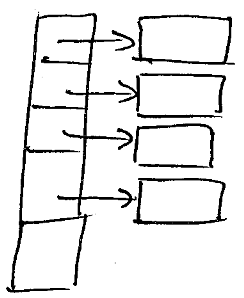
⇾ lo último en moverse dentro de la legalidad

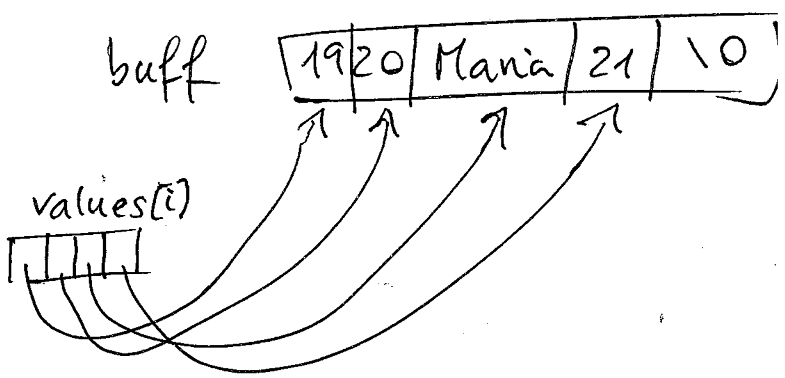
→ lo que rompe el bucle

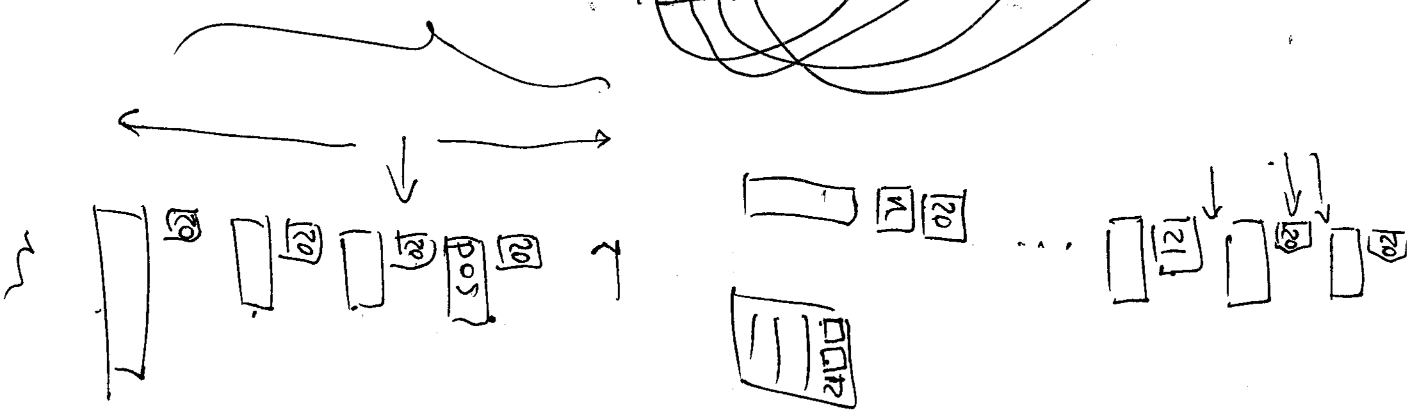
CONCLUSIÓN: P siempre acaba en el índice donde


P
realloc
P
vertical
P
NEW_RECORD


key
n-ptrs
i




buff
19
20
Mania
21
\0
values[i]



PRIMARY INDEX

SSN	OFFSET
15	249
25	152
35	227
45	100
55	181
65	129
75	206

SECONDARY INDEXES

GENDER	SSN
F	55
F	65
F	75
M	15
M	25
M	35
M	45

DEPT	SSN
C	55
C	75
HR	25
I	45
I	65
M	15
M	45

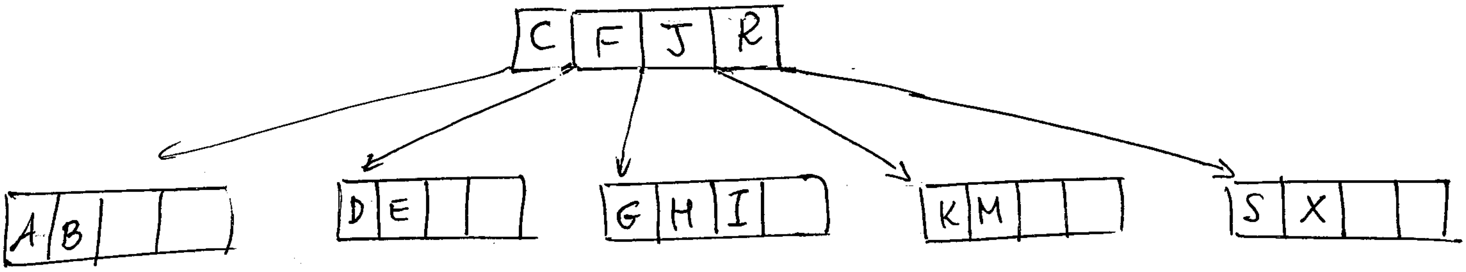
¿Female employees that work in department "I"?

$A \leftarrow \sigma_{DEPT='I'} I_{DEPT}$ $\quad$ $B \leftarrow \sigma_{GENDER='F'} I_{GENDER}$

$RES \leftarrow A \cap B$

CREATE M- TREE $m = 5$

A G F B K D H M J E S I R X C L N T U P



	MIN	MAX
0	1	1
1	2	N
2	$2.N$	N^2
3	$2.N^2$	N^3
4	$2.N^3$	N^4
⋮		
h	$2.N^{(h-1)}$	N^h

$N = \frac{M}{2}$

1. As an example let us compotu the amount of time it will take to:

- Read 1 Mbyte (10^6 bytes)
- From a 7200 RPM drive
- With a 8ms average seek time
- that has 500 sectors per track
- Each sector has 512 bytes
- Assume all sectors are consecutive

- Number of sectors needed to store the file?

  $10^6 / 512 \simeq 1954$

- Time to read one track?

  $\frac{7200 \text{ rpm}}{60} = 120 \text{rps}$; $\frac{1}{120} = 0'00835 \text{ s.} = 8'35 \text{ ms}$

- Time to read one sector?

  $\frac{8'35 \text{ ms/track}}{500 \text{ sectors/track}} = 16'6 \, \mu s/\text{sector}$

2. Imagine that we have a 2MB file. This file is stored in a magnetic hard disk withe the following characteristi

- 40 sectors per track
- 8 sectors per cluster
- 512 per sector
- Average seek time = 9'5 ms
- Spin speed = 5400 rpm

How long will it take to read the whole file in the worst and best cases?

- Number of sectors: $2.10^6/512 = 3907$ sectors
- Number of tracks: $3907/40 = 98$ tracks

24. RELATIONAL CALCULUS (ex. 2, 5, 6)

2) a) $\{FN.Follower\_nick \mid Follows(FN) \wedge \exists FN2 (Follows(FN2) \wedge followed\_nick = \text{'Luis'} \wedge$
$\exists FN3 (Follows(FN3) \wedge followed\_nick = \text{'Maria'} \wedge FN2.follower\_nick = FN3.follower\_nick))$

b) $\{FN.Follower\_nick \mid Follows(FN) \wedge \exists FN2 (Follows(FN2) \wedge FN.follower\_nick = FN2.followed\_ni$
$\wedge \exists FN3 (Follows(FN3) \wedge FN2.followernick = FN3.followed\_nick \wedge FN3.followed.nick =$
$= \text{'Nicola'}))\}$

c) $\{FN.Followed\_nick \mid Follows(FN) \wedge \exists FN2 (Follows(FN2) \wedge follower\_nick = \text{'Luis'} \wedge$
$\exists FN3 (Follows(FN3) \wedge follower\_nick = \text{'Maria'} \wedge FN2.followed\_nick = FN3.followed\_nick)$

$\{FN.Follower\_nick \mid Follows(FN) \wedge \exists FN2 (Follows(FN2) \wedge FN.followed\_nick = FN2.follower\_nick$
$\wedge \exists FN3 (Follows(FN3) \wedge FN2.followed\_nick = FN3.follower\_nick \wedge$
$\wedge FN3.follower\_nick = \text{'Nicola'}))\}$

5) a) $\{F.flight\text{-}id \mid Flight(F) \wedge \exists A (Airport(A) \wedge F.origin = A.code \wedge A.city = \text{'Paris'})$

b) $\{F.flight\_id \mid Flight(F) \wedge \exists A (Airport(A) \wedge F.origin = A.code \wedge A.city = \text{'Madrid'}$
$\wedge \exists A2 (Airport(A2) \wedge F.destination = A2.code \wedge A2.city = \text{'Paris'} \wedge$
$F.departure\_time = \text{'12:00:00'}))\}$

c) $\{P.name,\ B.date \mid Passengers(P) \wedge Bookings(B)\ \exists F, \exists A(Flights(F) \wedge Airport(A)$
$F.origin = A.code \wedge A.city = \text{'London'} \wedge \exists A2(Airport(A2) \wedge F.destination = A2.code$
$\wedge A2.city = \text{'Paris'}) \wedge F.flight\_id = B.flight\_id) \wedge P.dni = B.dni\}$

d) $\{P.name \mid Passengers(P) \wedge \exists A, \exists B, \exists F(Flights(F) \wedge Bookings(B) \wedge Airport(A) \wedge$
$B.dni = P.dni \wedge F.flight\_id = B.flight\_id \wedge \exists A2(Airport(A2) \wedge$
$((F.origin = A.code \wedge A.city = \text{'London'} \wedge F.destination = A2.code \wedge A2.city = \text{'Paris'})$
$OR\ (F.origin = A.code \wedge A.city = \text{'Paris'} \wedge F.destination = A2.code \wedge A2.city = \text{'London}$
$))\}$

e) $\{P.name \mid Passengers(P) \wedge \exists B, \exists F(Flights(F) \wedge Bookings(B) \wedge Airport(A) \wedge$
$B.dni = P.dni \wedge F.flight\_id = B.flight\_id \wedge \exists F2, \exists B2(Flights(F2) \wedge$
$Bookings(B2) \wedge B2.dni = P.dni \wedge B2.flight\_id = F2.flight\_id \wedge$
$\wedge F2.origin = F.destination \wedge F2.destination = F.origin \wedge$
$\wedge((B1.date - B2.date) < 24)))\}$

6 a) $\{A.name \mid Airlanes(A) \wedge \exists F, \exists Ap(Flights(F) \wedge Airports(Ap) \wedge$
$A.abbreviation = F.abbr\_airlane \wedge F.origin = Ap.code \wedge Ap.city = \text{'London'})\}$

c) $\{F.flight\_id \mid Flights(F) \wedge \neg\exists B(Bookings(B) \wedge B.flight\_id =$
$F.flight\_id)\}$

d) $\{Al.name \mid Airlanes(Al) \wedge \forall F\,(Flights(F) \wedge F.abbrairlane = Al\_abbreviation \wedge$
$\wedge \exists Ap\,(Airport(Ap) \wedge ((F.origin = Ap.code \wedge Ap.city != \text{'Madrid'}) \text{ OR } (F.destination = Ap.code$
$\wedge Ap.city != \text{'Madrid'}))))\}$

$\boxed{25.}$ RELATIONAL ÁLGEBRA (ex. 2, 5, 6)

2 a)

$RES1 \leftarrow \Pi_{follower\_nick}(\sigma_{followed\_nick = \text{'Luis'}}(Follows))$

$RES2 \leftarrow \Pi_{follower\_nick}(\sigma_{followed\_nick = \text{'Maria'}}(Follows))$

$RES\_FINAL \leftarrow RES1 \cap RES2$

b)

$RES1 \leftarrow \Pi_{follower\_nick}(\sigma_{followed\_nick = \text{'Nicola'}}(Follows))$

$RES\_FINAL \leftarrow \Pi_{follower\_nick}(\sigma_{followed\_nick}(RES1))$

c) a) $RES1 \leftarrow \Pi_{followed\_nick}(\sigma_{follower\_nick = \text{'Luis'}}(Follows))$

$RES2 \leftarrow \Pi_{followed\_nick}(\sigma_{follower\_nick = \text{'Maria'}}(Follows))$

$RES\_FINAL \leftarrow RES1 \cap RES2$

b) $RES1 \leftarrow \Pi_{followed\_nick}(\sigma_{follower\_nick = \text{'Nicola'}}(Follows))$

$RES\_FINAL \leftarrow \Pi_{followed\_nick}(\sigma_{follower\_nick}(RES1))$

5 a) $$\Pi_{flight\_id}\left(\sigma_{\delta}\ (Flights \times Airport)\right)$$

$\delta = (origin = code \wedge city = 'Paris')$

b) $$A \leftarrow \left(\sigma_{origin=code \wedge city='Madrid'}\ (Flights, Airports)\right)$$

$$B \leftarrow \Pi_{flight\_id}\left(\sigma_{destination=code \wedge city='Paris' \wedge departure\_time='12:00:00'}(A)\right)$$

c) $$A \leftarrow \Pi_{flight\_id}\left(\sigma_{origin=code \wedge city='London'}\ (Flights \times Airport)\right)$$

$$B \leftarrow \Pi_{flight\_id}\left(\sigma_{destination=code \wedge city\,!='Paris'}\ (Flights \times Airport)\right)$$

$$C \leftarrow A - B$$

$$RES\_FINAL \leftarrow \Pi_{passengers\_name,\ date}\left(\sigma_{C.flight\_id = Bookings.flight\_id \wedge Passengers.dni = Bookings.dni}\ (C \times Passengers \times Bookings)\right)$$

d) $$A \leftarrow \Pi_{flight\_id}\left(\sigma_{origin=code \wedge city='London'}\ (Flights \times Airport)\right)$$

$$B \leftarrow \Pi_{flight\_id}\left(\sigma_{destination=code \wedge city\,!=Paris}\ (Flights \times Airport)\right)$$

$$C \leftarrow \Pi_{flight\_id}\left(\sigma_{origin=code \wedge city='Paris'}\ (Flights \times Airport)\right)$$

$$D \leftarrow \Pi_{flight\_id}\left(\sigma_{destination=code \wedge city\,!='London'}\ (Flights \times Airport)\right)$$

$$RES1 \leftarrow ((A-B) \cup (C-D))$$

$$RES\_FINAL \leftarrow \Pi_{passengers.name}\left(\sigma_{RES1.flight\_id = Bookings.flight\_id \wedge pass.dni = book.dni}\ (RES1 \times Bookings \times Passeng\right.$$

⑥

a) $RES1 \leftarrow \Pi_{airlanes.name}(Airlanes)$

$RES2 \leftarrow \Pi_{airlanes.name}\left(\sigma_{abbreviation=abbr\_airlane \wedge origin=code \wedge city='London'}(Airlanes \times Flights \times Airport)\right)$

$RESFINAL \leftarrow RES1 - RES2$

c) $RES1 \leftarrow \Pi_{flight\_id}(Flights)$

$RES2 \leftarrow \Pi_{flight\_id}(Bookings)$

$RES\_FINAL \leftarrow RES1 - RES2$

d) $A \leftarrow \Pi_{airlanes.name}(Airlanes)$

$B \leftarrow \Pi_{airlanes.name}\left(\sigma_{origin=code \wedge city \neq 'Madrid' \wedge abbr\_airlane = abbreviation}(Airlanes \times Flights \times Airport)\right)$

$C \leftarrow \Pi_{airlanes.name}\left(\sigma_{destination=code \wedge city \neq 'Madrid' \wedge abbr\_airlane = abbreviation}(Airlanes \times Flights \times Airport)\right)$

$RES\_FINAL \leftarrow A - (B \cup C)$

28.

a)

PASSENGER

DNI	Name
123	María
456	Pedro
789	Isabel

RESERVATION

DNi	Number	Date	Price
123	345	20-12-10	170
456	345	03-11-10	190

b) Passenger's and reservation's tables completed

c)

PASSENGER

DNi	Name
123	María
456	Pedro
789	Isabel

RESERVATION

Number
165
345
321
345

d)

PASSENGER

Name
Pedro
María

e)

PASSENGER

Name
Isabel

31.

a)

RESERVATION

DNi
789
789

b)

PASSENGER		RESERVATION			
DNi	Name	DNi	Number	Date	Price
123	María	789	165	07-01-11	210
123	María	123	345	20-12-10	170
123	María	789	321	15-12-10	250
123	María	456	345	03-11-10	190

c)

NAME	DNi	NUMBER	DATE	PRICE
~~Isabel~~	~~789~~	~~165~~	~~07-01-11~~	~~210~~
María	123	345	20-12-10	170
~~Isabel~~	~~789~~	~~321~~	~~15-12-10~~	~~250~~
~~Pedro~~	~~456~~	~~345~~	~~03-11-10~~	~~190~~

d)

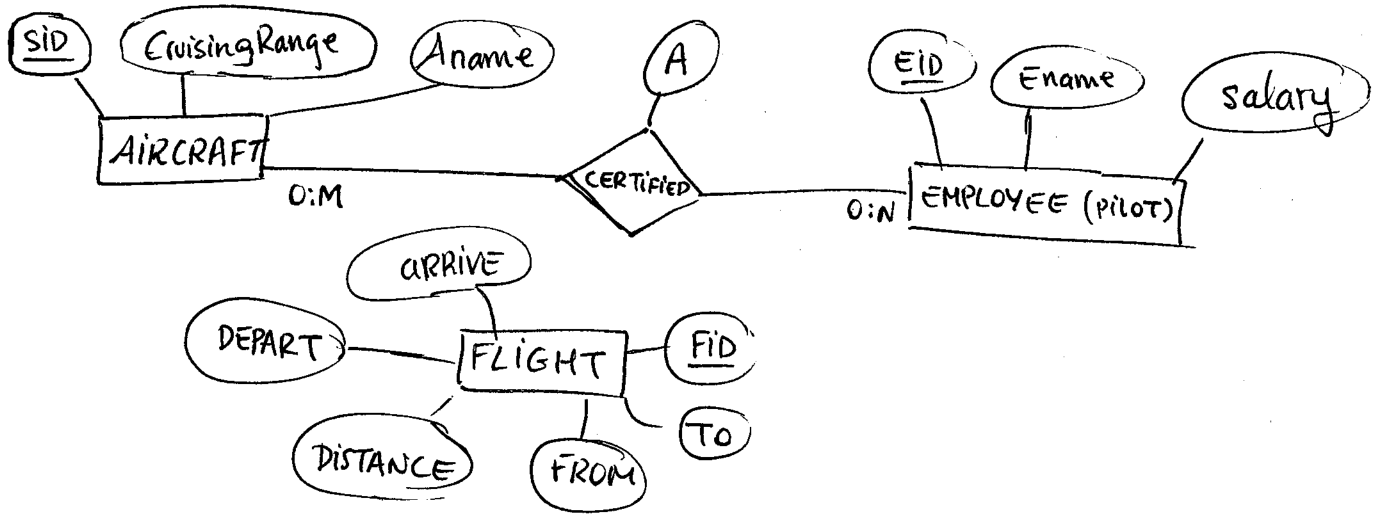
NAME	DNi	NUMBER	DATE	PRICE
María	123	345	20-12-10	170

e)

NAME	PRECIO	DNi
Pedro	190	456

f)

CONVERT E-R DIAGRAM TO RELATIONAL MODEL



FLIGHT (FID, From, to, Distance, Arrive, Depart)

AIRCRAFT (AID, Aname, CruisingRange)

EMPLOYEE (EID, Ename, Salary)

CERTIFIED (AID, EID, A)

- Pilot that can fly from New York to Boston.

```
SELECT Ename
FROM   Flight, Aircraft, Employee, Certified
WHERE  From = 'New York' AND to = 'Boston' AND distance ≤ cruising-range
       AND aircraft.aid = certified.aid AND certified.eid = employee.eid ;
```

Find PIDs of the most expensive parts supplied by the supplier named 'ACME'.

$\boxed{\text{SQL}}$

```
CREATE VIEW Max-cost AS
SELECT Max(cost) AS cost
FROM Catalog, Supplier
WHERE Sname = 'ACME' AND supplier.sID = catalog.sID;

SELECT PID
FROM Max-cost, Catalog
WHERE Max-cost.cost = Catalog.cost AND Sname = 'ACME';
```

$\boxed{\text{ALG}}$

Max-cost ⟵ [1st query]

[2nd query]

$\boxed{\text{CALCULUS}}$

$$\{ C1.PID \mid CATALOG(C1) \wedge \exists S(SUPPLIER(S) \wedge (S.NAME = 'ACME') \wedge$$
$$\wedge (C.SID = S.SID) \wedge \neg \exists C2 (CATALOG(C2) \wedge (C2.SID = S.SID) \wedge$$
$$(C2.COST > C1.COST))) \}$$

~~{ S.SID / SUPPLIER(S) ∃C,P (CATALOG(C) ∧ PARTS(P) ∧~~

~~S.SID = C.SID ∧ C.PID = P.PID ∧ ¬∃ P2 (PARTS(P2) ∧~~

~~P2.PID = C.PID ∧~~

Pairs of suppliers (S1, S2) that sell the same part (P1)
but S1.P1cost > S2.P1.cost

SQL

```
SELECT C1.SID , C2.SID
FROM Catalog C1 , Catalog C2
WHERE C1.COST > C2.COST AND C1.PID = C2.PID;
```

ÁLGEBRA

$C1 \leftarrow CATALOG$

$C2 \leftarrow CATALOG$

$$\pi_{C1.SID,\ C2.SID}\ \sigma_{C1.COST > C2.COST\ \wedge\ C1.PID = C2.PID}\ C1 \times C2$$

CALCULUS

$$\left\{ \begin{matrix} C1.SID \\ C2.SID \end{matrix} \;/\; CATALOG(C1) \wedge CATALOG(C2) \left((C1.PID = C2.PID) \wedge (C1.COST > C2.COST) \right) \right\}$$

SELECT Sid

FROM Suppliers

WHERE ~~sid~~ suppliers.sid = catalog.sid AND catalog.pid =

parts.pid AND ~~pa~~ color = 'red"

INTERSECTION

[

AND color = 'green' ;

we are talking about this DB.

SUPPLIERS (<u>SID</u>, sname, Address)

PARTS (<u>PID</u>, pname, color)

CATALOG (<u>SID</u>, <u>PID</u>, COST)

In calculus:

$$\{ S.NAME \mid CATALOG(C1) \wedge \exists P1 (PART(P1) \wedge (P1.COLOR = 'RED') \wedge (P1.PID = C1.PID)) \wedge \exists C2 (CATALOG(C2) \wedge (C1.SID = C2.SID) \wedge \exists P2 (PART(P2) \wedge (P2.COLOR = 'GREEN') \wedge P2.PID = C2.PID) \wedge \exists S (SUPPLIERS(S)) \wedge (C1.SID = S.SID)) \}$$

REPASO NORMAL FORMS

R(b, o, a, q, c, d)

b = broker

o = broker's office

a = share

q = n^er of shares

c = client

d = divident

$a \rightarrow d$

$c \rightarrow b$

$\{c, a\} \rightarrow q$

$b \rightarrow o$

R.1 (b, o, <u>a</u>, q, <u>c</u>, d) 1st NF

- → R1.1 (<u>a</u>, <u>c</u>, q)
- → R1.2 (<u>a</u>, d)
- → R1.3 (<u>c</u>, b, o)

2nd NF

- → R1.3.1 (<u>c</u>, b)
- → R1.3.2 (<u>b</u>, o)

3rd NF & BCNF

R(A, B, C, D, E, F, G)

B → A, C, D, E, F, G

E → F, G

R.1. (A, <u>B</u>, C, D, E, F, G) 1st NF

- → R.1.1. (A, <u>B</u>, C, D, E)
- → R.1.2. (<u>E</u>, F, G)

2nd NF

R(A, B, C, D)

A, B → C, D

C → B

Candidates-Key
A, B
A, C

→ R1(<u>A</u>, <u>B</u>, C, D) 1st, 2nd, 3rd

- → R1.1(<u>A</u>, <u>B</u>, D)
- → R1.2 (<u>C</u>, B)

BCNF