

MÈTODES EMMAGATZEMATGE

Informació es guarda en fitxer en disc i el SO auvedaix a ella i però és el SGBD el que sap com interpretar-ho.

Fitxer $\xrightarrow{\text{conté}}$ Extensions $\xrightarrow{\text{conté}}$ Pàgines. #Matrïos cas.

Fitxer

Unitat global que el SO gestiona. Es conjunt extensions.

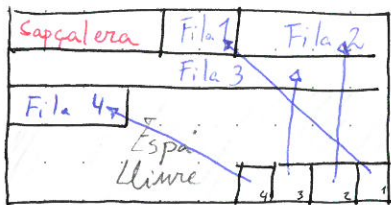
Extensió

Agrupació de $m \in \mathbb{N}$ pàgines contigües per gestionar l'espai en la BD.

Si una taula creix i necessita més espai, el SGBD pot assignar una nova extensió (que pot ser un "bloc" de 10 pag.) per ficar les noves dades. Redueix fragmentació.

Pàgina (Física)

Unitat més petita d'organització/transport de dades. Longitud fixa. 8KB.



• Vector @: Conté puntes per cada fila

• Capçalera: # No importa

∇ ∇ Realment apunten inici fila.

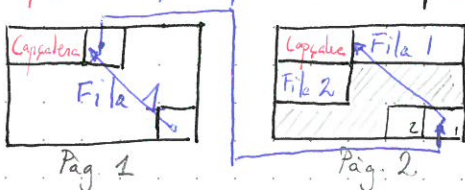
Fila X



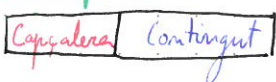
• Capçalera: Longitud total de la fila, Id. taula,

Si una fila no cap en la mateixa pàgina. \Rightarrow

Posterior a la Capçalera hi ha un punter al punter d'inici de la pàgina on estigui el restant.



Camp X



• Capçalera: Dia \leftrightarrow NULL (0) NOT NULL i longitud del camp.

∇ Si contingut fix i NOT NULL \Rightarrow NO té capçalera.

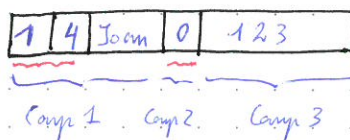
Exemple

CREATE TABLE T (

Camp 1 VARCHAR(10),

Camp 2 VARCHAR(10),

Camp 3 CHAR(10) NOT NULL
);



Fixet que no hi ha ~ en Camp 3 i x. és NULL no ocupa espai.

INSERT INTO T VALUES ("Joan", NULL, "123");

TABLESPACE

Component que implementa abstracció entre dades lògiques i físiques.

Dintre Espai Virtual (EV) hi ha relació 1:1 entre pàgines Virtuals ↔ Físiques.

"Següència pàg. virtuals que representen forma lògica l'espai emmagatzematge".



Permet guardar diferents taules en mateix fitxer o mateixes taules diferents fitxers.

⇒ Nivell Virtual: Les @ són propietat de SGBD.

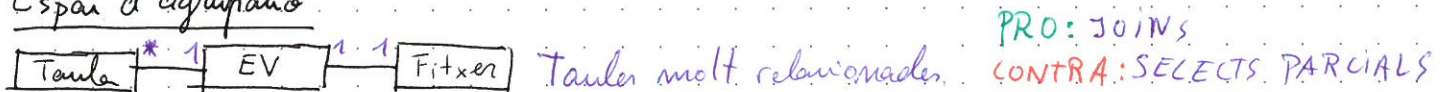
⇒ Nivell Físic: Les @ són propietat del SO.

Tipus

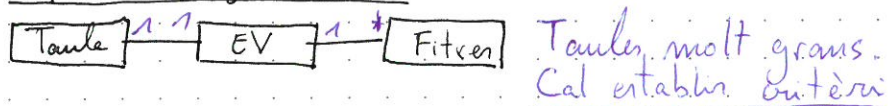
Espai de Taules



Espai d'agrupació



Espai de fragmentació



Espai d'objectes grans

S'emmagatzemen separat. Pàg. de diferent mida i úniques per l'objecte.

Mètodes d'Accés

Tipus	Modalitat	Descripció	Cost	Exemple
Per Posició	Directe	Obtindre pàg. demanada pel num.	1 pag.	INSERT INTO t VALUES
	Seqüencial	Obtindre pag. en ordre num.	N pag.	SELECT * FROM t
Per Valor	Directe	Obtindre fila amb valor atribut	1 pag.	* WHERE x = y
	Seqüencial	Obtindre files ordenades valor	N pag.	* ORDER BY x

Arbres B⁺

En comptes d'accedir a tots els Blocks on està la informació guardada (Pg és costosa) simplement accedim als Blocks on estan guardats els índexs i allí hi haurà un ptr fins la dada que volem saber.

Estructura

L'arbre té un valor d que és: "ordre de magnitud" i diu que $\begin{matrix} \nearrow 2d+1 \text{ ptrs} \\ \searrow 2d \text{ valors} \end{matrix}$

Això sig. que internament són:  Arbre B⁺ amb $d=2$.

- Nodes \rightarrow Interns: Serveixen per guar
- \rightarrow Root: Serveixen per guar
- \rightarrow Fulla: Contenen totes les PK i els seus RID corresponent.

$$2d(\text{mida} - \text{valor} + \text{RID}) + 2(\text{mida} - \text{ptr}) \quad \# 2 \text{ per apuntar } \leftarrow \boxed{\text{ptr}} \rightarrow$$

Normalment la mida del node $\leq 4 \text{KB} = 4096 \text{B}$

⚠️ Cada Node s'emmagatzema en una pàgina.

Normes

- Cada node ha d'estar ocupat almenys 50%. # Mínim d valors (0's $\cdot 2d \cdot d$)
- Tots nodes fulla mateix nivell # permet fer més fàcil càlculs.

Agrupats v.s. No Agrupats

- Agrupats (Cluster): Dades s'organitzen amb mateix ordre en el disc que en

Bones per quan agafem la taula. (Ordenats per la PK)
dades ordenades per la PK.

Norms podem tindre un índex agrupat per taula pg norms 1 ordre.

Per mantenir ordre es deixa espai entre pàg. > Si aquest es supera, les noves pàg. es queden guardades al final > Si hi ha moltes guardades al final \Rightarrow Reorganitza.

- No Agrupats: Cada entrada té un ptr a la fila que conté la PK.

Implementació de B⁺

Per Valor

Seqüencial ($a \geq X$)

- **Agrupat**: Simplement busquem el primer que compleix i recorrem el fitxer.
Es per això que importa saber quant elements i quants caben per pàg.
- **No Agrupats**: Donat que fulles estan ordenades, és important saber el cost fins arribem fulles + Quantes fulles hem de recórrer. C
Com que cada fulla té més d'1 valor, també hem de sumar 6 quant.

Directe ($a = X$)

- **A/NA, NVR**: Com que no hi ha valors repetits sabem que només 1 accés a Block.
No importa com estigui ordenat.
- **A, VR**: Ara podem haver valors repetits, però com que estan agrupats, simplement hem de saber quants blocs hem d'accedir i l'algada fins fulles.
- **NA, VR**: No estan agrupats i poden haver diversos. Estem mateix cas que Seq. NA.

Diversos Valors

Directe

- **Intersecció**: Analitzem de forma indep. cada atribut i després fem la suma.
Es important saber quants elements hi ha en comú. \Rightarrow N° pàgines mínim.
Si hi ha molts valors amb atrib1 i pos atrib2 \Rightarrow Poc eficient.
- **Multi-Atribut**: Primer s'ordenen pel primer atribut i es guarda "en una llista" els elements.
D'aquesta llista s'ordenen pel segon atribut, ...

Formulari ∇ Si Root en mem $\Rightarrow -1$.

Per Valor	Seqüencial	Agrupat	$h + \lceil \frac{R(a \geq X)}{f} \rceil$	Diversos Valors	Directe
		No Agrupat	$h + F + R(a \geq X) $	Intersecció	$(h_a + F_a) + (h_b + F_b) + R(a \wedge b)$
	Directe	A/NA, NVR	$h + 1$	Multi-Atrib	$h + F + \lceil \frac{R(\text{cond})}{f} \rceil$
		NA, VR	$h + F + R(a = X) $		
		A, VR	$h + \lceil \frac{R(a = X)}{f} \rceil$		$h = \lceil \log_{ptr} (\text{total valors}) \rceil$