

El nivel Interno

Índice Tema 3

iEl nivel Interno.....	1
Introducción.....	2
Etapas de acceso a la base de datos.....	2
Gestor de Disco.....	3
Gestor de Ficheros.....	3
Agrupamiento.....	4
Interactivos.....	4
Entrearchivos.....	4
Paginas y Ficheros.....	4
Administración de Paginas.....	4
Administración del Registro.....	7
Indexación.....	8
Índices densos.....	8
Índices no densos.....	9
Direccionamiento directo y Hashing.....	10
Hashing.....	10
Técnicas de direccionamiento directo.....	10
Transformación de plegamiento.....	10
Cuadrados Centrales.....	10
Métodos de Convergencia (Mas utilizado).....	11
Desplazamiento.....	11
Cadenas de Punteros.....	11
Técnicas de Compresión.....	11
Técnica Diferencial.....	11
Jerárquica.....	12
Código Huffman.....	12

Introducción

Finalidad del DBA en el nivel interno

- Minimizar el numero de accesos a disco.
- Poder Trabajar con distintas estructuras de almacenamiento.

Etapas de acceso a la base de datos

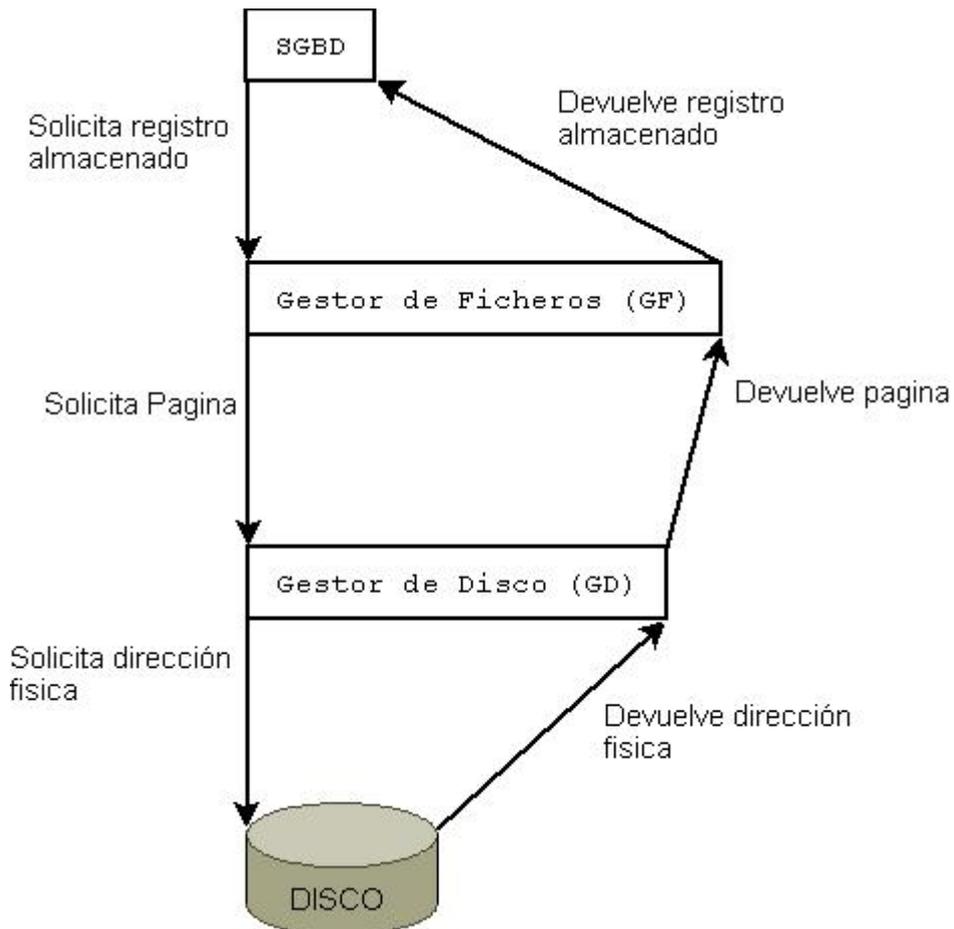


Ilustración 1: diagrama de las etapas de acceso a la base de datos

Gestor de Disco

El gestor de disco es un elemento del sistema operativo subyacente que se encarga de realizar todas las operaciones de entrada salida físicas.

Permite que el gestor de ficheros perciba la base de datos como conjuntos de paginas, cada conjunto de paginas esta compuesto por grupos de paginas de tamaño fijo donde cada grupo se identifica de manera única a su vez cada pagina posee un identificador único.

$$CP1 \cap CP2 \cap CP3 = \emptyset$$

Los conjuntos de paginas son disjuntos (el identificador de cada pagina es único para todo el disco), el gestor de discos de encarga de mantener una correspondencia entre cada pagina y su dirección física en disco.

Es necesario saber el numero de paginas libres que quedan en disco para permitir la introducción de nuevos datos, estas paginas forman lo que se denomina el “conjunto de paginas del espacio libre”, el gestor de disco se encarga de asignar o liberar paginas según las peticiones del gestor de ficheros estas pueden ser las siguientes.

1. Leer la pagina P del conjunto de paginas C.
2. Reemplazar la pagina P del conjunto de paginas C
3. Añadir la pagina P al conjunto de paginas C
4. Eliminar la pagina P del conjunto de paginas C

Gestor de Ficheros

El gestor de ficheros utiliza al gestor de disco para que el SGBD perciba el disco como un conjunto de archivos almacenados.

Cada conjunto de paginas puede contener uno o mas tipos de archivo almacenados donde cada uno de ellos tendrá un identificador único en el conjunto de paginas y a su vez cada registro almacenado tendrá un identificador único **RID** (register identificador) dentro de cada archivo almacenado, generalmente el gestor de ficheros forma parte del Sistema Operativo.

Las operaciones que el SGBD puede solicitar al gestor de ficheros son las siguientes.

- Leer el registro almacenado R del archivo almacenado A.
- Reemplazar el registro almacenado R del archivo almacenado A.
- Añadir un nuevo registro almacenado al archivo almacenado y devolver el nuevo identificador.
- Eliminar el registro R del archivo almacenado A.
- Crear un nuevo archivo almacenado A.
- Eliminar el archivo almacenado A.

Agrupamiento

El agrupamiento se utiliza para que los registros que se utilizan juntos estén en la misma pagina existen 2 tipos

Interactivos

En una misma pagina o en paginas que estén cercanas físicamente estén agrupados registros de un fichero que estén juntos.

Entrearchivos

Se utiliza cuando las aplicaciones requieren con frecuencia acceso a varios archivos almacenados.

Paginas y Ficheros

Administración de Paginas

Consiste en permitir que el gestor de ficheros se olvide de todas las entradas y salidas físicas y piense en paginas

La pagina 0 es una pagina especial donde se guardan los puntos de inicio y de fin de los diferentes registros almacenados.

0
Registros S ->Pagina de Comienzo
Registros P ->Pagina de Comienzo
Registros SX/PY ->Pagina de Comienzo
Registros Libres ->Pagina de Comienzo

0	¹ S1 ²	² S2 ³	³ S3 ⁴	⁴ S4 ⁵	⁵ S5 ^{FIN}	⁶ P1 ⁷
⁷ P2 ⁸	⁸ P3 ⁹	⁹ P4 ¹⁰	¹⁰ P5 ¹¹	¹¹ P6 ^{FIN}	¹² S1/P1 ¹³	¹³ S1/P2 ¹⁴
¹⁴ S1/P3 ¹⁵	¹⁵ S2/P1 ¹⁶	¹⁶ S2/P4 ¹⁷	¹⁷ S3/P1 ¹⁸	¹⁸ S3/P3 ¹⁹	¹⁹ S3/P5 ^{FIN}	²⁰ ²¹
²¹ ²²	²² ²³	²³ ²⁴	²⁴ ²⁵	²⁵ ²⁶	²⁶ ²⁷	²⁷ ²⁸
²⁸ ²⁹	²⁹ ³⁰	³⁰ ³¹	³¹ ³²	³² ³³	³³ ³⁴	³⁴ ^{FIN}

Insertar un nuevo proveedor S6

0	¹ S1 ²	² S2 ³	³ S3 ⁴	⁴ S4 ⁵	⁵ S5 ²⁰	⁶ P1 ⁷
⁷ P2 ⁸	⁸ P3 ⁹	⁹ P4 ¹⁰	¹⁰ P5 ¹¹	¹¹ P6 ^{FIN}	¹² S1/P1 ¹³	¹³ S1/P2 ¹⁴
¹⁴ S1/P3 ¹⁵	¹⁵ S2/P1 ¹⁶	¹⁶ S2/P4 ¹⁷	¹⁷ S3/P1 ¹⁸	¹⁸ S3/P3 ¹⁹	¹⁹ S3/P5 ^{FIN}	²⁰ S6 ^{FIN}
21 22	22 23	23 24	24 25	25 26	26 27	27 28
28 29	29 30	30 31	31 32	32 33	33 34	34 FIN

Borrar S2

0	¹ S1 ³	² 21	³ S3 ⁴	⁴ S4 ⁵	⁵ S5 ²⁰	⁶ P1 ⁷
⁷ P2 ⁸	⁸ P3 ⁹	⁹ P4 ¹⁰	¹⁰ P5 ¹¹	¹¹ P6 ^{FIN}	¹² S1/P1 ¹³	¹³ S1/P2 ¹⁴
¹⁴ S1/P3 ¹⁵	¹⁵ S2/P1 ¹⁶	¹⁶ S2/P4 ¹⁷	¹⁷ S3/P1 ¹⁸	¹⁸ S3/P3 ¹⁹	¹⁹ S3/P5 ^{FIN}	²⁰ S6 ^{FIN}
21 22	22 23	23 24	24 25	25 26	26 27	27 28
28 29	29 30	30 31	31 32	32 33	33 34	34 FIN

Insertar P7

0	¹ S1 ³	² P7 ^{FIN}	³ S3 ⁴	⁴ S4 ⁵	⁵ S5 ²⁰	⁶ P1 ⁷
⁷ P2 ⁸	⁸ P3 ⁹	⁹ P4 ¹⁰	¹⁰ P5 ¹¹	¹¹ P6 ²	¹² S1/P1 ¹³	¹³ S1/P2 ¹⁴
¹⁴ S1/P3 ¹⁵	¹⁵ S2/P1 ¹⁶	¹⁶ S2/P4 ¹⁷	¹⁷ S3/P1 ¹⁸	¹⁸ S3/P3 ¹⁹	¹⁹ S3/P5 ^{FIN}	²⁰ S6 ^{FIN}
21 22	22 23	23 24	24 25	25 26	26 27	27 28
28 29	29 30	30 31	31 32	32 33	33 34	34 FIN

Borrar S4

0	¹ S1 ³	² P7 ^{FIN}	³ S3 ⁵	⁴ 21	⁵ S5 ²⁰	⁶ P1 ⁷
⁷ P2 ⁸	⁸ P3 ⁹	⁹ P4 ¹⁰	¹⁰ P5 ¹¹	¹¹ P6 ²	¹² S1/P1 ¹³	¹³ S1/P2 ¹⁴
¹⁴ S1/P3 ¹⁵	¹⁵ S2/P1 ¹⁶	¹⁶ S2/P4 ¹⁷	¹⁷ S3/P1 ¹⁸	¹⁸ S3/P3 ¹⁹	¹⁹ S3/P5 ^{FIN}	²⁰ S6 ^{FIN}
21 22	22 23	23 24	24 25	25 26	26 27	27 28
28 29	29 30	30 31	31 32	32 33	33 34	34 FIN

Las misiones del gestor de disco a nivel de disco con el administrador de disco son las siguientes:

- Mantener el orden lógico de la paginas.
- Mantener información general de los conjuntos de paginas (*pagina 0*).
- Optimizar las entradas y salidas a nivel de paginas.

Administración del Registro

*Suponemos que tenemos mas de un registro almacenado por pagina.

Supongamos que un registro tiene la siguiente estructura:

S1	<Registro>
S2	<Registro>
S3	<Registro>

Relocalización: Ordenar los registros dentro de una pagina para tener el espacio libre al final.

RID	La pagina en la que esta	Dirección o posición dentro de la pagina
-----	--------------------------	--

El RID o identificador de registro permanece Invariable. Los registros dentro de la pagina permanecen agrupados al principio de la pagina sin existir huecos intermedios al producirse un borrado se suben todos los registros posteriores al registro borrado si estos existieran.

Para mantener el RID invariable lo que se hace es modificar la estructura de una pagina añadiendo al final una zona de índices

S1	<Registro>		
S2	<Registro>		
S3	<Registro>		
*S2	*S1	*S3	<-- posición del registro (aquí habrá tantos campos como registros en pagina)

Lo que se hace es que el campo del RID sea un puntero a la posición dentro de la página

RID	La pagina en que esta el registro	Posición en el fondo de la pagina
-----	-----------------------------------	-----------------------------------

Una vez resuelto el problema de como mantener el RID invariable ahora necesitamos encontrarlo de la forma mas rápida posible teniendo dos opciones la Indexación y las Técnicas de acceso

directo y Hashing

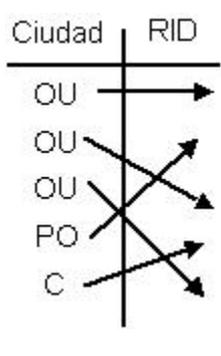
Indexación

La indexación permite la obtención de registros de una manera mas rápida, minimizando de esta forma los accesos a disco y la lectura de paginas. Existen 2 técnicas

- Índices densos.
- Índices no densos.

Índices densos

Una tabla índice de índices no densos es un fichero formado por dos partes, la primera de ellas es un valor semántico y la segunda son la dirección del registro que posee ese valor semántico



<i>S/F</i>	<i>NomS</i>	<i>Estado</i>	<i>Ciudad</i>
S1	X	10	OU
S2	X	20	PO
S3	Y	10	OU
S4	Z	30	C
S5	W	10	OU

Ventajas: accesos secuenciales.

Inconvenientes: excesos de información.

Índices no densos

Los índices no densos generan un fichero que tiene dos partes, en la primera se guarda el mayor valor de una pagina y en la segunda parte se guarda la pagina en la que esta ese registro.

S#	Pag	S#	Nums	Estado	Ciudad	Pagina
S2	→	S1	X	10	OU	P1
		S2	X	20	PO	
S4	→	S3	Y	10	OU	P2
		S4	Z	30	CO	
S5	→	S5	W	10	OU	P3

Los arboles de los índices no densos son los arboles B.

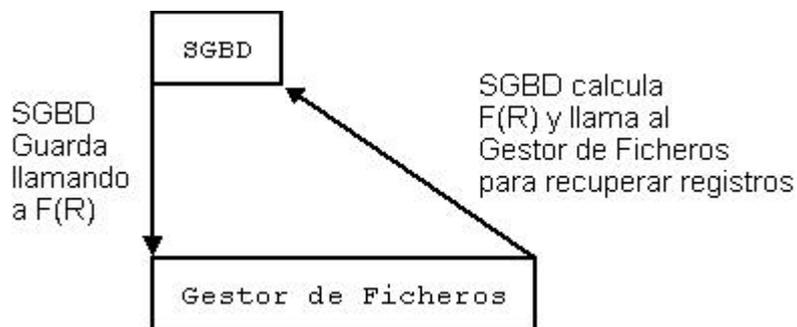
Direccionamiento directo y Hashing

Hashing

El direccionamiento directo hashing permite obtener acceso directo a N registros almacenados mediante la obtención de un valor dado para un cierto campo.

Se da en 3 pasos

1. Cada registro almacenado se coloca en la base de datos en un RID que se calcula como una función F denominada algoritmo de direccionamiento $f(R) \rightarrow RID$.
2. El sistema de gestión de bases de datos calcula la dirección y le ordena al gestor de ficheros que lo grave en esa posición para localizar el registro a partir de la dirección.
3. El SGBD realiza el mismo calculo y ordena al gestor de ficheros que lea el registro en la posición calculada.



Inconvenientes:

- Sinónimos para dos valores diferentes de la clave obtengo la misma dirección, por lo tanto tengo que crear zonas de desbordamiento otra solución es dejar huecos en las paginas.
- Huecos para los huecos la única solución es diseñar un buen algoritmo de direccionamiento, técnicas de direccionamiento aleatorio.

Técnicas de direccionamiento directo

Transformación de plegamiento

Consiste en transformar la clave a binario, pasarla a binario con un tamaño de 8 bits y sumar los cuatro primeros con los cuatro últimos el resultado es el RID.

Cuadrados Centrales

Elevar la clave al cuadrado y cogemos los dígitos centrales necesarios para obtener el RID.

Métodos de Convergencia (Mas utilizado)

El RID se obtiene por $RID = \frac{(K + B)}{(Mod(N))}$ <-- ¿Quién es Quien?

Desplazamiento

Se utiliza cuando la clave es una cadena numérica grande y consiste en dividir esos números y sumarlo hasta obtener el tamaño de RID que nos interesa.

Cadenas de Punteros

Se utilizan cuando queremos trabajar con atributos cuyos valores se repiten, son eficientes su ese atributo tiene pocos valores y se repite mucho.

Inconvenientes: Tenemos que recorrer la lista

Ventajas: Facilidad de inserción o borrado

(para mi que esto esta incompleto)

Técnicas de Compresión

Las técnicas de comprensión se utilizan para ahorrar memoria masiva se basan en el siguiente principio. Los valores de los datos casi nunca son aleatorios sino que existe un grado de parecidos, siguen una secuencia entre ellos.

Técnica Diferencial

Consiste en reemplazar cada valor del dato individual por alguna representación del valor que existe entre él y el valor que le precede.

<i>Normal</i>	<i>Comprimido</i>
Marta	Marta
Marti	5-i
Martín	6-n
Martinez	8-ez

Jerárquica

Los registros se agrupan según una llave que tiene duplicados. Entonces se realiza una compresión dentro del archivo obtenido.

Interarchivo	Ourense P1 ... S2
Entrearchivo	S1 ... P1 ... P3 ...

Código Huffman

Se basa en la codificación interna de los datos, consiste en cambiar esa codificación teniendo en cuenta la aparición de esos datos.

E	30%	1
A	30%	01
D	20%	001
C	10%	0001
B	5%	0001

Se utiliza en la búsqueda y recuperación de la información